

Travail de Bachelor 2011

## Filière Informatique de gestion

### Etude CRM/ERP existants et adaptation pour Axianet.ch



Etudiante : Milène Fauquex

Professeur : Alexandre Cotting

## I. Préface

Le travail de bachelor a pour vocation d'achever les trois ans de formation nécessaires à l'obtention du titre d'informaticien de gestion.

Le rôle d'un informaticien de gestion est d'analyser les besoins d'une entreprise et d'y apporter des solutions adaptées. Cette analyse inclut évidemment de comprendre le mode de fonctionnement de l'entreprise, les contraintes et les particularités de son activité et également les besoins et souhaits des utilisateurs.

A l'heure actuelle, un système d'information est la plupart du temps informatisé. Véritable centre névralgique de l'entreprise, il permet de traiter, stocker et communiquer toutes les informations nécessaires à son bon fonctionnement. Un système d'information peut donc jouer un rôle capital dans le succès d'une entreprise.

Les deux notions évoquées ci-dessus décrivent ce à quoi nous avons été sensibilisés et formés durant ces trois dernières années.

Ce travail de bachelor applique ces notions à un cas réel : une entreprise qui désire optimiser une partie de son système d'information. De l'analyse de l'entreprise à la livraison d'un prototype fonctionnel, en passant par la comparaison des besoins exprimés avec les caractéristiques des produits existants. Ce travail a impliqué beaucoup de communication, d'analyse, de conseils, et de prises de décisions. Ce rapport décrit les différentes étapes qui ont été nécessaires à son bon déroulement.

## II. Remerciements

Je souhaite remercier cordialement toutes les personnes qui m'ont permises de réaliser ce travail de bachelor dans les meilleures conditions.

Un remerciement particulier à :

M. Alexandre Cotting, mon professeur responsable, pour m'avoir encadrée et suivie tout au long de ce travail. Merci pour sa disponibilité et les conseils prodigués.

M. Benoît Eberlé, d'Axianet.ch, pour son implication dans le projet ainsi que pour ses réponses à mes nombreuses questions concernant le fonctionnement de l'entreprise et les aspects logiques et techniques de l'application.

M. Yves Bochatay, d'Axianet.ch, pour sa disponibilité et les explications fournies au sujet du Framework Yii.

Enfin, merci à ma maman et à M. Steven Moix pour leurs conseils et la relecture de ce rapport.

### III. Déclaration sur l'honneur

Je déclare, par ce document, que j'ai effectué le travail de bachelor ci-annexé seul, sans autre aide que celles dûment signalées dans les références, et que je n'ai utilisé que les sources expressément mentionnées. Je ne donnerai aucune copie de ce rapport à un tiers sans l'autorisation conjointe du RF et du professeur chargé du suivi du travail de bachelor, y compris au partenaire de recherche appliquée avec lequel j'ai collaboré, à l'exception des personnes qui m'ont fourni les principales informations nécessaires à la rédaction de ce travail et que je cite ci-après : Benoît Eberlé et Yves Bochatay, d'Axianet.ch.

Lieu et date :

Signature :

---

## IV. Résumé

Ce travail de bachelor a été réalisé avec la collaboration de l'entreprise Axianet.ch. Cette entreprise souhaitait obtenir une application de gestion interne de ses activités. L'objectif de ce travail a été de fournir à Axianet.ch une solution qui réponde au mieux à ses besoins. Pour ce faire, plusieurs étapes ont été nécessaires. Elles sont décrites ci-après :

### **Analyse de l'entreprise et de ses besoins**

Afin de proposer une solution pertinente, il a tout d'abord été nécessaire d'effectuer une analyse des activités de l'entreprise. J'ai eu la chance de pouvoir collaborer ponctuellement avec Axianet.ch sur certains de leurs projets par le passé, le fonctionnement de l'entreprise ne m'était donc pas totalement inconnu. De plus, le cahier des charges fourni en préambule était particulièrement détaillé, ce qui a facilité l'analyse.

Sur la base de ces connaissances et après discussions avec l'entreprise, une liste de fonctionnalités à implémenter dans le cadre du travail de bachelor a pu être établie. Cette phase d'analyse a également permis d'identifier certains processus métiers utiles dans le fonctionnement de la solution, ainsi que des cas d'utilisation.

### **Analyse des outils existants et choix d'une solution**

L'analyse préalable de l'entreprise a permis de mettre en évidence les points nécessaires à l'analyse des outils existants et d'en définir la portée. Les critères retenus pour l'étude d'un outil existant sont les suivants : l'outil doit être de type CRM/ERP, gratuit et open-source, orienté web, écrit en PHP et doit pouvoir être déployé sur une infrastructure de type LAMP (Linux Apache MySQL PHP).

Sur la base de ces pré-requis, neuf outils ont été retenus et analysés. La liste des fonctionnalités définie dans l'analyse précédente a été détaillée et chacun des outils a été confronté à cette liste et classifié selon une liste de critères pour savoir dans quelle mesure il remplissait la fonctionnalité.

Cette analyse a permis d'identifier deux outils qui répondaient aux critères de manière satisfaisante : SugarCRM et vtigerCRM. Ces outils ont ensuite été déployés et configurés de manière à ce que l'entreprise puisse les tester.

### **Développement d'un prototype sur la base de l'analyse précédente**

Suite aux tests effectués sur les outils mis en place et après discussion, il a été décidé avec l'entreprise que la solution serait finalement développée depuis zéro à l'aide du Framework PHP Yii, déjà utilisé, et donc connu, par Axianet.ch. Pour développer le prototype, l'approche agile a été choisie : un product backlog a été défini et les différentes user stories ont été priorisées avec l'aide du client. Ensuite, le développement a été réalisé durant quatre itérations, avec pour chaque itération, le choix des fonctionnalités à implémenter en accord avec le client.

## V. Structure du document

Ce document débute avec une introduction décrivant le cadre du travail de bachelor et ses objectifs. Vient ensuite une analyse de l'entreprise : son mode de fonctionnement, ses processus et le choix des fonctionnalités retenues, dans le cadre de ce travail, qui répondent au mieux aux besoins actuels de l'entreprise.

La partie suivante décrit les technologies choisies pour la réalisation du prototype. Le document détaille ensuite l'analyse complète des outils CRM/ERP étudiés dans le cadre du projet et précise la solution à implémenter.

Les détails de cette solution sont exposés dans la partie d'après. Enfin ce rapport se termine sur la description de la gestion du projet et un bilan général contenant notamment les difficultés rencontrées, les améliorations futures et un point de vue personnel sur l'ensemble du travail.

## Sommaire

<b>I.</b>	<b>Préface.....</b>	<b>I</b>
<b>II.</b>	<b>Remerciements .....</b>	<b>II</b>
<b>III.</b>	<b>Déclaration sur l'honneur .....</b>	<b>III</b>
<b>IV.</b>	<b>Résumé.....</b>	<b>IV</b>
<b>V.</b>	<b>Structure du document .....</b>	<b>V</b>
	<b>Sommaire.....</b>	<b>1</b>
<b>1</b>	<b>Introduction .....</b>	<b>5</b>
1.1	Contexte .....	5
1.2	Objectifs.....	5
1.2.1	Objectifs principaux.....	5
1.2.2	Objectifs secondaires .....	5
<b>2</b>	<b>Analyse de l'entreprise.....</b>	<b>6</b>
2.1	Présentation .....	6
2.2	Cahier des charges initial .....	6
2.3	Fonctionnalités choisies .....	7
2.4	Description des processus.....	7
2.4.1	Processus 1 : Entrer un travail (facturation) .....	8
2.4.2	Processus 2 : Entrer les heures des techniciens (non facturées) .....	9
2.5	Cas d'utilisation.....	10
2.5.1	Acteurs .....	10
2.5.2	Cas d'utilisation : utilisateurs .....	10
2.5.3	Cas d'utilisation : clients .....	11
2.5.4	Cas d'utilisation : travaux et heures.....	12
2.5.5	Cas d'utilisation « Facturer élément ».....	13
2.5.6	Cas d'utilisation « Entrer les heures ».....	13
2.5.7	Cas d'utilisation : secteurs, statuts et offres .....	14
<b>3</b>	<b>Choix technologiques.....</b>	<b>15</b>
3.1	Open source et orienté web .....	15
3.2	PHP.....	15
3.3	MySQL .....	15
3.4	Architecture.....	16
<b>4</b>	<b>Analyse des outils existants.....</b>	<b>17</b>
4.1	Choix des outils à analyser .....	17

---

4.2	Liste d'outils CRM/ERP analysés .....	17
4.3	Critères d'analyses et de comparaison.....	17
4.4	Résultat des analyses.....	18
4.5	Conclusion.....	18
4.6	Test et adaptation .....	18
4.6.1	Feedback de l'entreprise .....	19
4.7	Choix de la solution à implémenter .....	19
<b>5</b>	<b>Solution sur mesure .....</b>	<b>20</b>
5.1	Framework imposé : Yii.....	21
5.1.1	Présentation.....	21
5.1.2	Fonctionnalités .....	21
5.2	Modélisation des données .....	25
5.2.1	Clients.....	25
5.2.2	Offres .....	25
5.2.3	Statuts_offres.....	26
5.2.4	Packs_heures .....	26
5.2.5	Travaux.....	26
5.2.6	Statuts.....	26
5.2.7	Secteurs .....	26
5.2.8	Elements_factures .....	26
5.2.9	Unites.....	26
5.2.10	Utilisateurs.....	26
5.2.11	Heures .....	27
5.3	Description de l'application .....	27
5.3.1	Interface et navigation.....	27
5.3.2	Login .....	28
5.3.3	Clients.....	29
5.3.4	Travaux.....	30
5.3.5	Eléments à facturer.....	31
5.3.6	Heures .....	33
5.3.7	Pack d'heures .....	34
5.3.8	Offres .....	35
5.3.9	Utilisateurs.....	35
<b>6</b>	<b>Gestion du projet.....</b>	<b>37</b>
6.1	Planification initiale .....	37



---

6.2	Approche agile.....	38
6.2.1	Product backlog.....	39
6.2.2	Définition de « done ».....	40
6.3	Itérations .....	40
6.4	Relation avec l'entreprise.....	41
<b>7</b>	<b>Synthèse.....</b>	<b>42</b>
7.1	Taux de complétion .....	42
7.1.1	De l'importance de l'analyse .....	43
7.2	Difficultés rencontrées.....	43
7.2.1	Analyse des outils.....	43
7.2.2	Prise de décision .....	43
7.2.3	Appréhension du framework .....	44
7.3	Améliorations .....	44
7.3.1	Implémentation de la gestion des rôles .....	44
7.3.2	Amélioration de l'ergonomie .....	45
7.3.3	Amélioration de la ligne graphique.....	45
7.4	Bilan personnel .....	45
<b>8</b>	<b>Glossaire.....</b>	<b>47</b>
<b>9</b>	<b>Bibliographie .....</b>	<b>48</b>
9.1	Analyse des outils CRM/ERP existants .....	48
9.2	Développement.....	49
9.3	Divers .....	49
9.4	Source des images.....	50
<b>10</b>	<b>Table des figures .....</b>	<b>51</b>
<b>11</b>	<b>Table des illustrations.....</b>	<b>52</b>
<b>12</b>	<b>Table des tableaux.....</b>	<b>52</b>
<b>13</b>	<b>Annexes.....</b>	<b>53</b>
13.1	Cahier des charges initial fourni par Axianet.ch.....	53
13.2	Cahier des charges du projet.....	54
13.3	Rapport de comparaison des outils CRM/ERP existants.....	55
13.4	Classification des outils CRM/ERP existants .....	56
13.5	Schéma des processus.....	57
13.6	Product Backlog.....	58
13.7	Backlogs des itérations.....	59
13.8	Planification initiale .....	60

---

13.9	Planifications des itérations .....	61
13.10	Rapports hebdomadaires des semaines 4 à 10 .....	62
13.11	Procès-verbaux des séances .....	63
13.12	Codes d'accès à l'application de test.....	64

# 1 Introduction

Cette première partie introduit le sujet en exposant tout d'abord le cadre du projet. Les objectifs principaux, puis les secondaires seront ensuite décrits.

## 1.1 Contexte

La plupart des entreprises possèdent un système de gestion informatisé. Il existe des solutions variées sur le marché actuel. Du simple CRM (Customer Relationship Management), aux ERP (Enterprise Resource Planning) plus complexes, difficile pour une entreprise qui souhaite s'équiper d'un tel système de trouver une solution adaptée à ses besoins.

## 1.2 Objectifs

### 1.2.1 Objectifs principaux

L'objectif principal de ce travail de bachelor était de fournir à la société Axianet.ch une solution de gestion interne de ses activités. Ce prototype doit en priorité répondre aux besoins des utilisateurs, être simple et rapide d'utilisation et être basé sur le langage PHP afin que l'équipe d'Axianet.ch puisse y apporter par la suite des modifications ou des fonctionnalités supplémentaires. Tout a été mis en œuvre afin que le prototype livré en fin de projet soit utilisable à un pourcentage le plus élevé et le plus rapidement possible.

C'est pourquoi une analyse des outils existants a été réalisée : il était important de savoir si une solution existante pouvait répondre immédiatement aux besoins de l'entreprise. Cette analyse constitue également un des objectifs principaux de ce projet. Elle a permis de se rendre compte au travers de l'étude de différents outils de ce qui était déjà disponible actuellement sur le marché des CRM/ERP. Elle a également contribué à cerner plus précisément les fonctionnalités demandées par l'entreprise et a ainsi grandement aidé à la proposition de solutions de qualité.

### 1.2.2 Objectifs secondaires

A mon sens, deux objectifs sous-jacents découlent de ce travail de bachelor.

Tout d'abord la collaboration avec une entreprise. Un des défis de ce projet était de coordonner les aspects académiques d'un travail de bachelor avec la réalité des besoins commerciaux d'une entreprise. Il s'agissait de définir et de proposer des solutions répondant aux besoins et aux contraintes de l'un comme de l'autre.

Un des autres objectifs secondaires était de travailler à l'aide d'un framework. Si l'un des outils étudiés dans la phase d'analyse avait été choisi, des contraintes de développement inhérentes à cet outil auraient de toute façon fait surface. Dans le cas d'une solution sur mesure, le développement à l'aide du framework PHP Yii a été demandé par l'entreprise.

## 2 Analyse de l'entreprise

Le chapitre suivant présente la manière dont s'est déroulée l'analyse de l'entreprise. Elle inclut tout d'abord une rapide présentation de l'entreprise et de la manière dont elle est gérée actuellement.



Illustration 1 - Logo d'Axianet.ch

La partie d'après décrit la manière dont le cahier des charges initial fourni par l'entreprise a été analysé pour découler sur la liste des fonctionnalités choisies dans le cadre du projet.

Sur cette base, on retrouve ensuite une analyse détaillée des processus auxquels le système devra répondre, et enfin une description des cas d'utilisations.

### 2.1 Présentation

Axianet.ch est une société valaisanne fondée en 2003 par Benoît Eberlé et Steven Moix. Elle est spécialisée dans le développement et l'hébergement de sites Internet ainsi que l'administration de serveurs. Inscrite au registre du commerce depuis 2005, elle s'est transformée en Sàrl début 2008.

En 2010, un nouvel associé, Yves Bochatay, a rejoint la société. Depuis lors, ses activités ne cessent de s'étendre.

Pour répondre à l'essor de ses activités, Axianet.ch avait besoin d'un système de gestion interne. Sur la base du cahier des charges initial décrit ci-après, une liste de fonctionnalités à fournir dans le cadre de ce travail de bachelor a été définie.

### 2.2 Cahier des charges initial

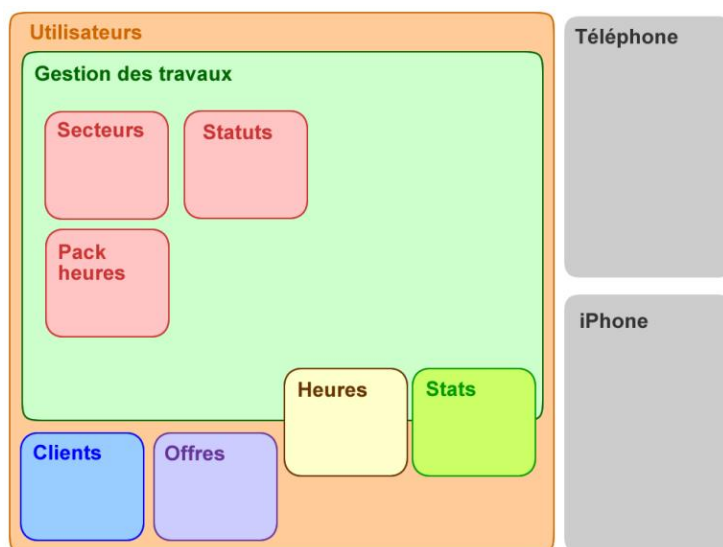


Figure 1 - Fonctionnalités initiales

En début du projet, un cahier des charges a été fourni par Axianet.ch. Ce cahier des charges comportait une liste exhaustive de tous les éléments désirés par la société pour son système de gestion. La figure ci-contre présente de manière visuelle ces fonctionnalités.

Une description détaillée des fonctionnalités est disponible dans le cahier des charges d'Axianet.ch fourni en annexe<sup>1</sup>.

Pour permettre d'adapter ce

<sup>1</sup> Voir annexe 13.1

cahier des charges à la portée du travail de bachelor, les fonctionnalités ont été analysées et en accord avec l'entreprise, certaines, jugées de trop grande envergure, comme l'application pour iPhone où l'intégration du serveur de téléphone Asterisk, ont été écartées. Il a toutefois été défini que la solution proposée permette aux collaborateurs d'Axianet.ch d'intégrer facilement ces autres fonctionnalités au système. Ce dernier point, comme nous le verrons plus tard, a été décisif quant au choix de la solution à implémenter.

La partie suivante décrit les fonctionnalités retenues dans le cadre du projet.

### 2.3 Fonctionnalités choisies

Les fonctionnalités retenues sont illustrées dans la figure 2. Il s'agit de la gestion des utilisateurs du système, de la gestion des travaux et des sous-fonctionnalités liées : secteurs, statuts et pack d'heures, la gestion des clients, une gestion simple des offres et enfin la gestion des heures des techniciens.

Ces fonctionnalités ont été considérées comme les plus prioritaires au fonctionnement de l'entreprise. La sélection a été volontairement restreinte afin que le prototype livré en fin de projet puisse être fonctionnel et utilisable à un taux le plus élevé possible. Le but étant de proposer une solution qui fasse épargner temps et argent à l'entreprise.

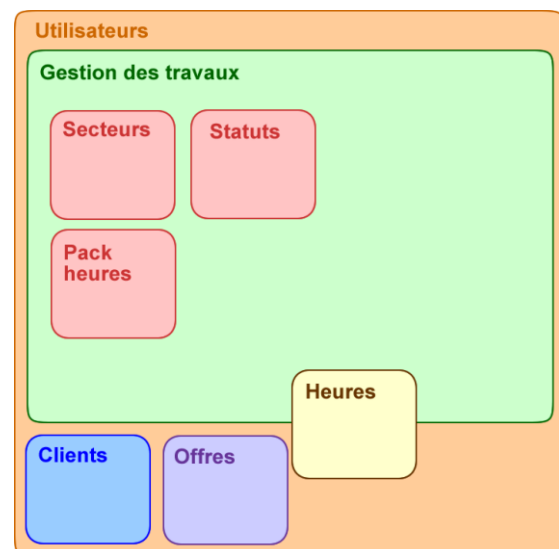


Figure 2 - Fonctionnalités choisies

### 2.4 Description des processus

En s'appuyant sur ce qui précède, deux processus principaux auxquels le système devra répondre ont pu être mis en lumière.

Le premier processus est l'entrée d'un travail en facturation. Il s'agit, pour un employé, d'entrer les éléments à facturer pour un client. Le deuxième processus est l'entrée des heures des employés. Ces heures ne sont pas facturées (elles sont par contre toujours liées à un client). Les deux processus sont donc indépendants. Le but de ce deuxième processus est de suivre, pour chaque employé, les heures réalisées jour par jour.

Ce chapitre décrit de manière détaillée chacun des processus.

### 2.4.1 Processus 1 : Entrer un travail (facturation)

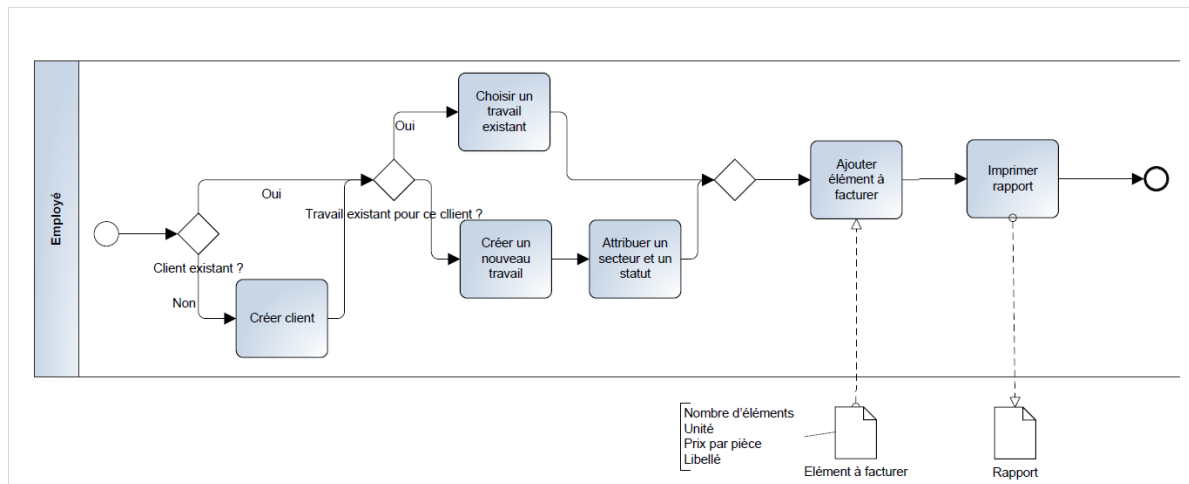


Figure 3 – Processus 1 : Entrer un travail (facturation)

#### Description générale

Comme décrit en début de chapitre, ce processus consiste à entrer les éléments à facturer à un client. La marche à suivre est la suivante :

- 1) Un employé sélectionne un client dans la liste des clients existants ou crée un nouveau client.
- 2) Si un travail est déjà en cours pour ce client, il le sélectionne, sinon il en crée un nouveau auquel il attribue un secteur et un statut.
- 3) Il entre ensuite l'élément à facturer pour ce client.
- 4) Un rapport peut éventuellement être imprimé pour le client ou la comptabilité.

#### Eléments d'entrée

L'élément d'entrée pour ce processus est l'élément à facturer. Celui-ci contient un nombre d'éléments, une unité (heure, kilomètre, forfait, pièce), le prix par pièce et un libellé. Cet élément pourra également être lié à un pack d'heures.

#### Eléments de sortie

L'élément de sortie pour ce processus est un rapport imprimable contenant un affichage résumé de tous les éléments facturés au client et l'en-tête de la société.

#### Acteurs

- **Employé** : sélectionne le client, crée le travail et ajoute l'élément à facturer
- **Client** : consulte et signe le rapport
- **Service de comptabilité** : consulte le rapport pour émettre une facture

## 2.4.2 Processus 2 : Entrer les heures des techniciens (non facturées)

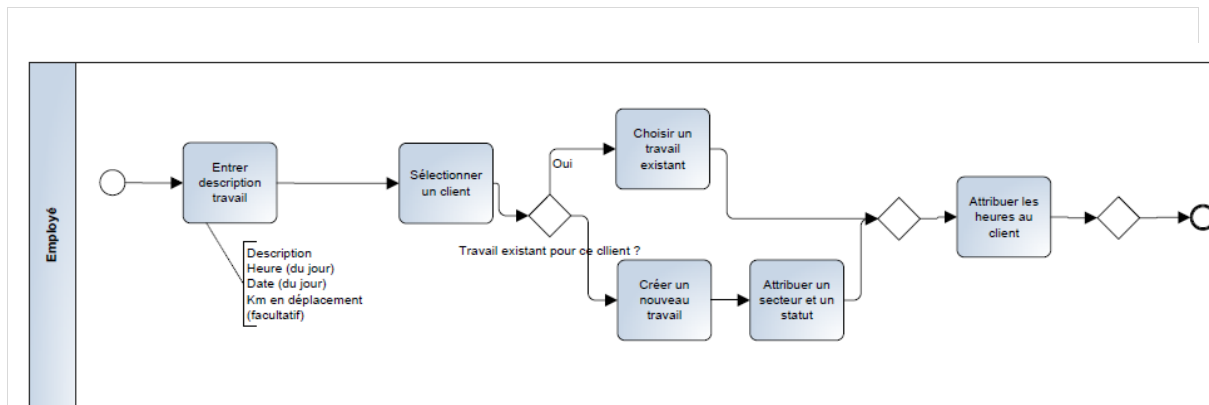


Figure 4 – Processus 2 : Entrer les heures des techniciens (non facturées)

### Description générale

Ce processus consiste en l'entrée des heures des employés. Chaque jour, les employés vont entrer les heures qu'ils réalisent durant la journée. Ce processus devra donc être simple et rapide afin que les employés n'y perdent pas de temps. Les étapes qui le composent sont les suivantes:

- 1) L'employé entre la description de son travail.
- 2) L'employé sélectionne le client correspondant.
- 3) Il pourra les attribuer à un travail déjà en cours pour ce client, ou en créer un nouveau.

### Éléments d'entrée

L'élément d'entrée pour ce processus est la description des heures de l'employé. Il contient la description du travail, l'heure et la date, et éventuellement un nombre de kilomètres en déplacement.

### Éléments de sortie

Un des éléments de sortie est un affichage par calendrier permettant de visualiser les heures entrées jour par jour avec l'affichage du total d'heures pour la journée et du total d'heures par mois.

L'autre élément est une page qui permet de lister tous les employés avec le nombre total d'heures de chaque mois et les kilomètres effectués. Cette page sera consultée par le service de comptabilité.

### Acteurs

- **Employé** : entre ses heures et les lie à un travail pour un client
- **Service de comptabilité** : consulte le total des heures afin de calculer les salaires et le remboursement des frais

## 2.5 Cas d'utilisation

Le chapitre suivant présente les différents acteurs et cas d'utilisation du système. Pour plus de clarté, les différents cas d'utilisation ont été séparés et classifiés par fonctionnalités du système.

### 2.5.1 Acteurs

On définit deux types principaux d'acteurs du système : les acteurs internes et les acteurs externes. Les acteurs internes sont les utilisateurs du système. Les acteurs externes quant à eux sont des acteurs qui auront des interactions avec certains éléments du système mais n'en seront pas utilisateurs à proprement parler.

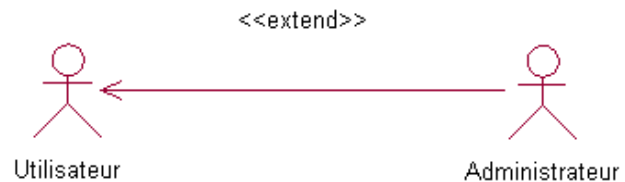


Figure 5 – Acteurs internes du système



Figure 6 – Acteurs externes du système

Dans les acteurs internes on retrouve l'utilisateur et l'administrateur. Un administrateur hérite d'un utilisateur.

Pour les acteurs externes, on retrouve le client et le service de comptabilité. Comme précisé plus haut, ces acteurs n'interagissent pas directement au cœur du système, mais seulement avec certains de ses éléments. On retrouvera plus tard les diagrammes de cas d'utilisation précisant ces interactions.

### 2.5.2 Cas d'utilisation : utilisateurs

**Acteurs impliqués** : utilisateur, administrateur

La figure 7 présente les actions réalisées par les utilisateurs et les administrateurs. Comme un administrateur hérite d'un utilisateur, il hérite également des actions qu'il réalise. Un utilisateur peut se loguer et se déloguer du système, il peut récupérer son mot de passe en cas d'oubli en fournissant son adresse e-mail. Il peut également éditer ses informations personnelles, comme son nom d'utilisateur ou son mot de passe.

L'administrateur, lui, peut gérer les utilisateurs, cela signifie en créer, en supprimer et également éditer les informations personnelles d'un utilisateur. Il peut aussi attribuer des droits d'accès aux différents utilisateurs.



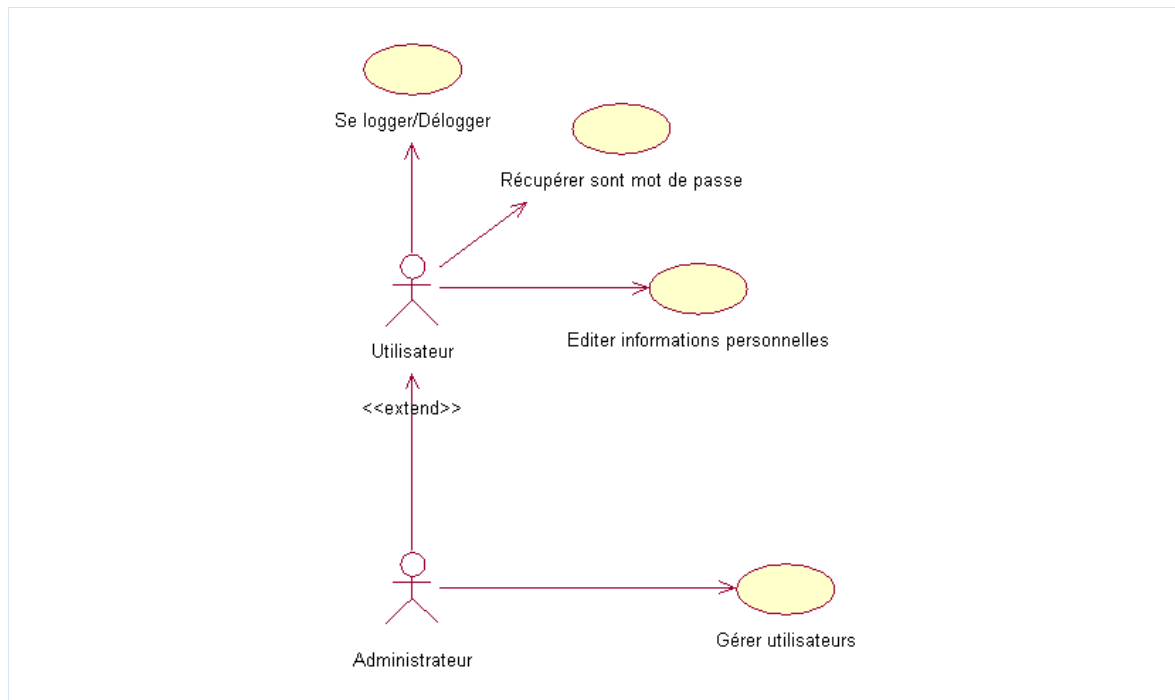


Figure 7 - cas d'utilisation : utilisateurs

### 2.5.3 Cas d'utilisation : clients

**Acteurs impliqués** : utilisateurs

Les interactions d'un utilisateur avec le module « client » sont présentées à la figure 8. Un utilisateur peut gérer une liste de clients, cela signifie en créer, en supprimer ou modifier les informations d'un client existant. Il peut également rechercher un client ou les trier par ordre alphabétique.

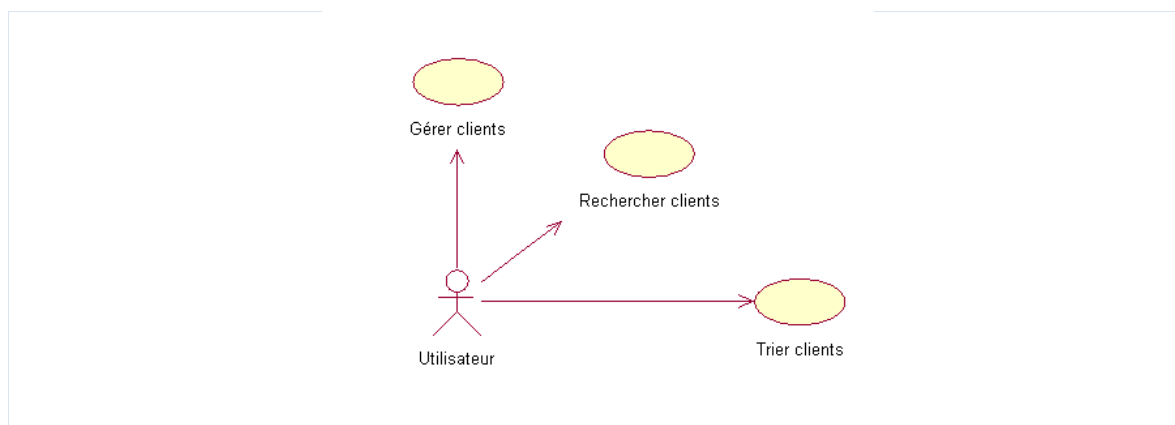


Figure 8 - cas d'utilisation : clients

### 2.5.4 Cas d'utilisation : travaux et heures

**Acteurs impliqués** : utilisateur, client, service comptabilité

A la figure 9, on retrouve deux modules où sont impliqués tous les acteurs.

Les cas d'utilisation en bleu représentent le module « travail ». Il s'agit du module dans lequel un utilisateur a la possibilité de facturer un élément pour un client. Le cas d'utilisation « Facturer élément » est illustré plus amplement figure 10. A propos d'un travail, un utilisateur peut le modifier, le supprimer, il peut également rechercher un travail dans la liste des travaux existants ou les trier selon leur statut. Enfin, un utilisateur peut visualiser tous les détails d'un travail. Cette visualisation sera également disponible sous forme d'un rapport (« Consulter rapport facturation »), qui sera aussi consultable par le service de comptabilité et par le client.

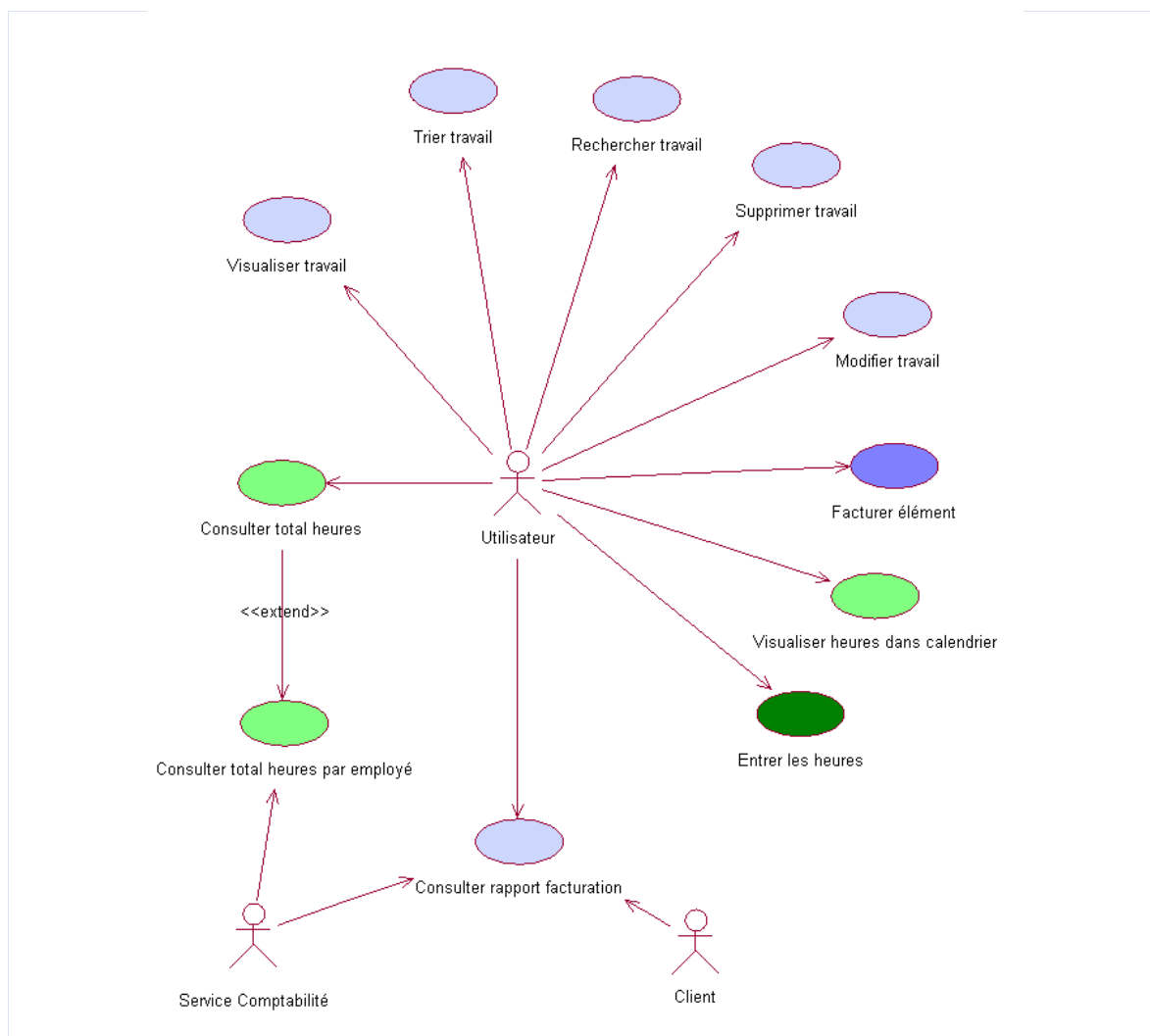


Figure 9 - cas d'utilisation : travaux et heures

En vert, on retrouve les cas d'utilisations concernant le module « heures ». C'est le module dans lequel les employés entrent les heures qu'ils réalisent jour par jour. Ces heures ne sont pas forcément facturées au client. Ainsi, les deux modules sont indépendants l'un de l'autre. A

nouveau ici, le cas d'utilisation « Entrer les heures » est décrit plus précisément à la figure 11. Un utilisateur peut visualiser les heures qu'il a entrées sous la forme d'un calendrier, et consulter le total de ses heures par jour et par mois. Le service de comptabilité, quant à lui, peut consulter ces mêmes totaux mais pour tous les employés.

### 2.5.5 Cas d'utilisation « Facturer élément »

Comme précisé plus haut, le diagramme suivant illustre plus précisément le cas d'utilisation « facturer élément ». Lorsqu'un utilisateur veut facturer un élément, cela implique qu'il doit sélectionner un travail existant pour un client donné. Il peut également créer un nouveau travail, dans ce cas, il sélectionne un client et attribue à ce travail un secteur et un statut. Ensuite, il ajoute l'élément à facturer. Celui-ci contient un nombre d'éléments, une unité (heure, kilomètre, forfait, pièce), le prix par pièce et un libellé. Cet élément pourra également être lié à un pack d'heures.

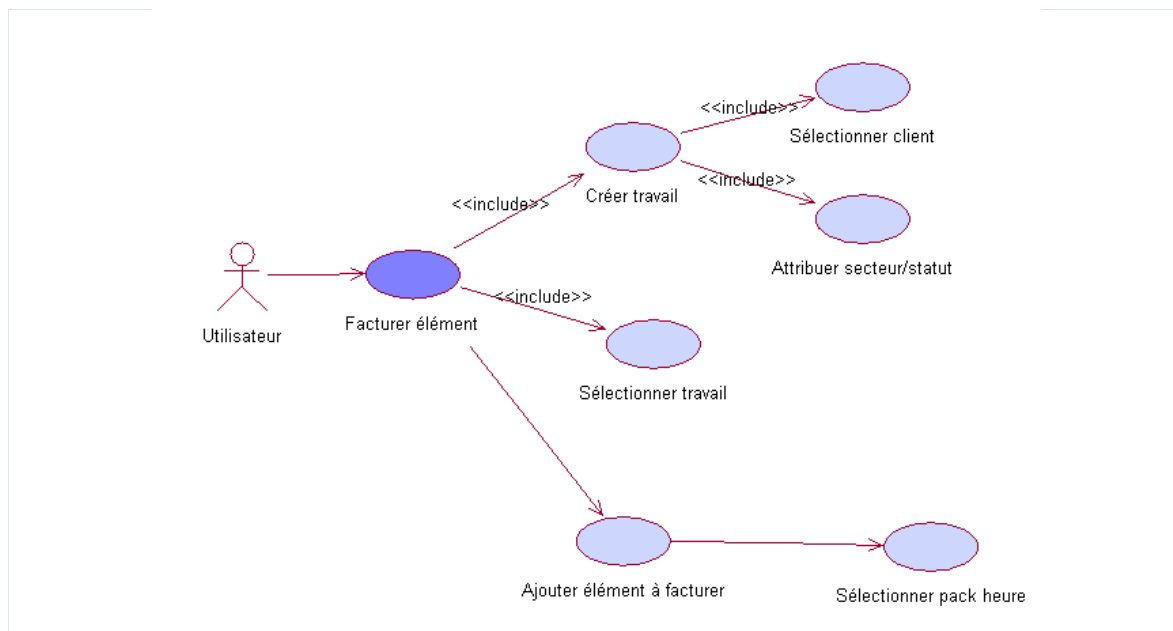


Figure 10 - cas d'utilisation « facturer élément »

### 2.5.6 Cas d'utilisation « Entrer les heures »

Comme pour la facturation, lorsqu'un employé entre ses heures cela implique qu'il doit sélectionner un travail existant pour un client donné. Il peut également créer un nouveau travail, dans ce cas, il sélectionne un client et attribue à ce travail un secteur et un statut.

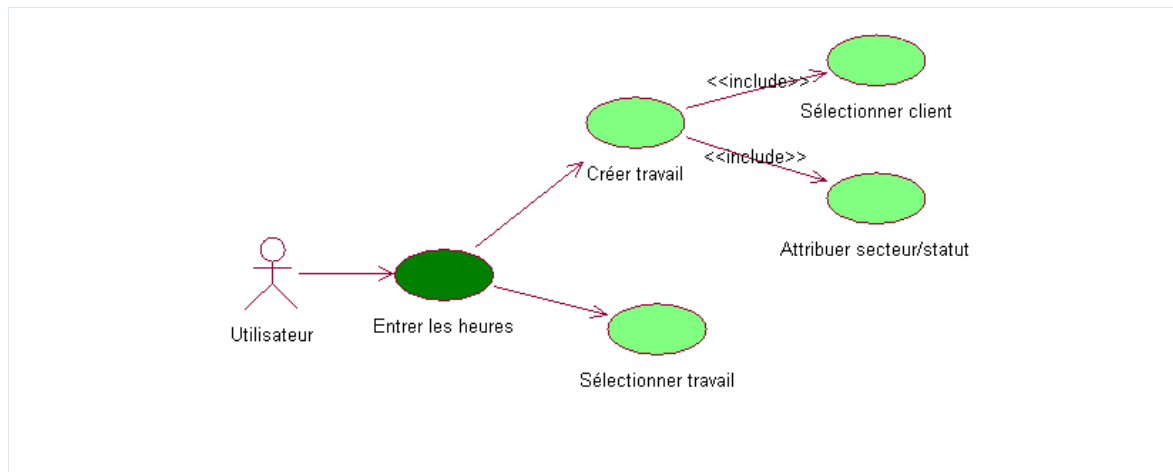


Figure 11 - cas d'utilisation « Entrer les heures »

### 2.5.7 Cas d'utilisation : secteurs, statuts et offres

**Acteurs impliqués** : utilisateur

Le diagramme suivant présente les cas d'utilisations pour les secteurs. Un utilisateur peut en créer, en supprimer ou en modifier un existant. De même pour les statuts et les offres. Pour les offres, de plus, il pourra visualiser les offres non terminées sous forme de liste et rechercher et trier parmi les offres existantes.

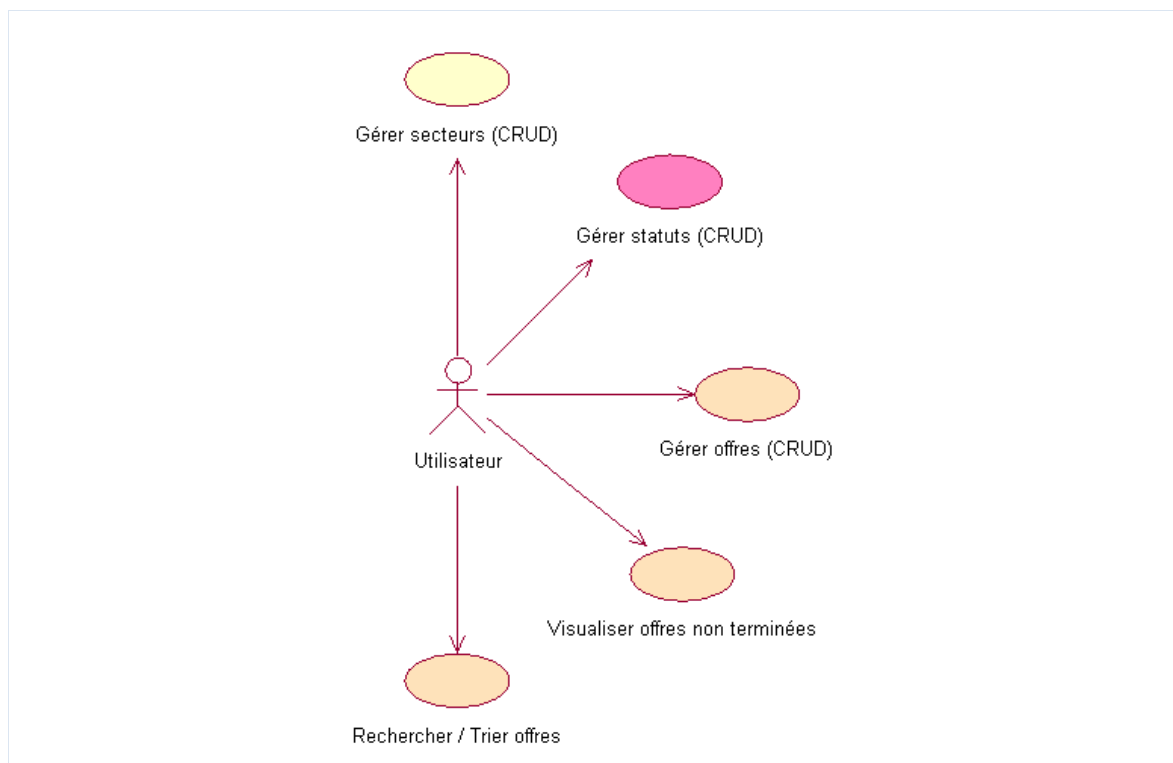


Figure 12 - cas d'utilisation : secteurs, statuts et offres

### 3 Choix technologiques

La section suivante a pour but de présenter brièvement les technologies choisies et surtout de justifier le pourquoi de ces choix. Certaines technologies sont considérées comme connues et seront donc seulement décrites de manière succincte.

#### 3.1 Open source et orienté web

Afin d'avoir un contrôle total sur le code de l'application et également de pouvoir effectuer des modifications ou des adaptations, la solution choisie se doit d'être open source. Si l'on se réfère à la définition de l'open source donnée par l'open source initiative, "open source ne signifie pas seulement l'accès au code source". En effet, on retrouve dans cette définition des aspects liés à la libre distribution, aux licences et à l'intégrité du code source de l'auteur. Dans notre cas, ce sont les aspects d'accès et de modification du code source qui nous intéressent.



Illustration 2 – Logo open source initiative

De plus, comme les collaborateurs d'Axianet.ch sont souvent en déplacement, ils veulent pouvoir facilement accéder à l'application, celle-ci sera donc orientée web.

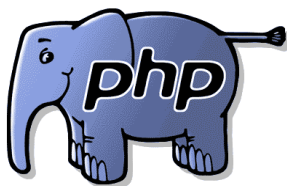


Illustration 3 - Logo PHP

#### 3.2 PHP

PHP est un langage de scripts libre largement utilisé pour le développement d'applications web dynamiques, il est en principe installé sur un serveur Apache.

Comme l'entreprise Axianet.ch est spécialisée notamment dans l'hébergement et le développement d'applications basées sur une infrastructure Linux, elle a demandé à ce que la solution fournie soit écrite en PHP, car c'est un langage qu'elle maîtrise et pratique depuis de nombreuses années. Cela permettrait de rendre la maintenance et la modification de l'application plus aisée.

#### 3.3 MySQL

MySQL figure parmi les systèmes de gestion de base de données (SGBD) les plus utilisés au monde. Comme PHP, MySQL fait partie du quatuor LAMP (Linux Apache MySQL PHP) et pour les mêmes raisons citées ci-dessus, il a été demandé par l'entreprise que l'application proposée repose sur une base de données de type MySQL. En effet, étant spécialisé dans la gestion de serveurs et l'hébergement Linux, Axianet.ch possède déjà toute l'infrastructure nécessaire au déploiement d'une telle solution.



Illustration 4 - Logo MySQL

### 3.4 Architecture

A terme, l'application sera déployée sur les serveurs d'Axianet.ch et accessible de manière sécurisée à l'interne de l'entreprise et via un VPN depuis l'extérieur. Les aspects liés au déploiement de l'application ne sont pas traités dans ce rapport.

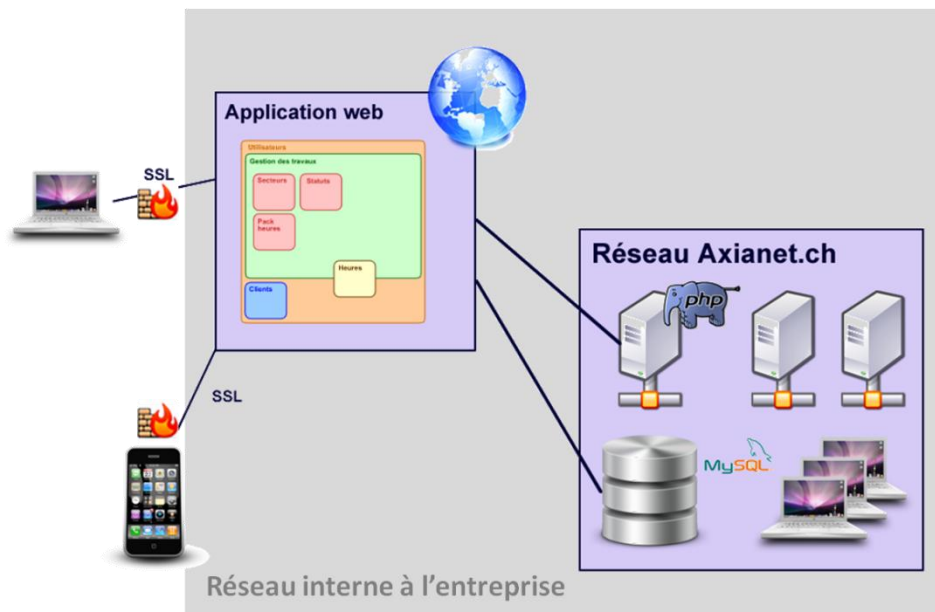


Figure 13 - Architecture

Afin de faciliter les interactions entre les divers participants du projet, un serveur et une base de données de développement seront mis à disposition par Axianet.ch. La version en développement ne contiendra pas de données sensibles et pourra donc être accédée via le web. En attendant que le système d'authentification de l'application soit totalement opérationnel, celle-ci sera protégée par un fichier .htaccess, qui permet de protéger l'accès à un dossier ou à un fichier avec un mot de passe.

## 4 Analyse des outils existants

La partie ci-après décrit de manière succincte le déroulement de l'analyse des outils CRM/ERP existants. Le rapport détaillé de cette analyse a été rédigé plus en amont du projet et est disponible en annexe<sup>2</sup>. Les points 4.1 à 4.5 de ce chapitre sont volontairement résumés car ils reprennent les éléments principaux découlant de ce rapport.

### 4.1 Choix des outils à analyser

Dans le chapitre 3, les choix technologiques et les raisons de ces choix ont été présentés. Les outils choisis pour être analysés devaient respecter ces technologies : être open-source, orienté web, écrit en PHP et tournant sur une infrastructure LAMP.

Durant la recherche, si un outil rencontrait ces critères, il était ajouté à la liste pour une analyse plus approfondie. Finalement, neuf outils ont été retenus pour la phase principale d'analyse. Ils sont listés ci-après.

### 4.2 Liste d'outils CRM/ERP analysés

Outils de type CRM (Customer Relationship Management)	Outils de type ERP (Enterprise Resource Planning)
SugarCRM Vtiger CRM XRMS CRM Dolibarr ERP/CRM EGGcrm	OpenERP Opentaps WebERP BlueERP

Tableau 1- Liste des outils analysés

### 4.3 Critères d'analyses et de comparaison

Le critère principal d'analyse d'un outil est de savoir s'il remplit ou non les fonctionnalités demandées. Pour ce faire, une liste détaillée des fonctionnalités a été établie, basées sur celles retenues lors de l'analyse du cahier des charges. Ensuite, les pertinences de chaque fonctionnalité dans l'outil étudié ont été classifiées selon les critères suivants:

- **N : Natif** – présent de manière native dans l'application et utilisable sans altération
- **A : Adaptable** – présent dans l'application mais besoin d'être adapté pour convenir à la fonctionnalité désirée (adaptation mineure)
- **C : Possibilité de créer/modifier** – absent de l'application mais possibilité de créer cette fonctionnalité ou de modifier une fonctionnalité actuelle pour convenir (adaptation majeure)
- **X : Inexistant**

<sup>2</sup> Voir annexe 13.3

Le déroulement de l'analyse et les critères sont décrits plus amplement dans le rapport de comparaison joint en annexe.

En outre, la classification détaillée telle que décrite ci-dessus est disponible dans le fichier Excel présent lui aussi en annexe<sup>3</sup>.

#### 4.4 Résultat des analyses

Globalement, l'analyse démontre que ce sont les outils de type CRM qui correspondent le mieux aux activités d'Axianet.ch. Les outils de type ERP impliquent des acteurs qu'on ne retrouve pas forcément dans l'activité de l'entreprise.

Alors que certaines fonctionnalités sont présentes dans tous les outils (gestion des clients et des utilisateurs), d'autres en revanche, plus spécifiques, ne sont pas du tout représentées (gestion des packs d'heures ou des offres).

Deux outils se démarquent des autres et apparaissent comment les plus complets: il s'agit de SugarCRM et vtigerCRM. Ces deux outils offrent, en plus des fonctionnalités demandées, une interface de personnalisation complète et une expérience utilisateur relativement ergonomique.



Illustration 5 - Logos SugarCRM et vtigerCRM

#### 4.5 Conclusion

Si les outils retenus dans cette analyse conviennent pour des fonctionnalités courantes, des écarts plus ou moins importants apparaissent lorsqu'on touche à des fonctionnalités plus spécifiques. Ces écarts impliqueraient évidemment des adaptations.

La question se pose de l'utilité d'adapter un outil existant, qui demandera dans tous les cas une adaptation et des compromis du point de vue de l'utilisabilité, alors qu'un outil sur mesure impliquerait également un développement, certes plus important, mais correspondrait de manière plus précise aux besoins de l'entreprise.

#### 4.6 Test et adaptation

Afin que l'entreprise puisse se faire une idée plus précise des outils choisis et avoir tous les éléments en mains pour prendre une décision quant à la solution à implémenter, les deux outils retenus : SugarCRM et vtigerCRM ont été installés et déployés sur un serveur de test fourni par Axianet.ch.

---

<sup>3</sup> Voir annexe 13.4



Une rapide adaptation de l'outil SugarCRM a été effectuée: un tri dans les fonctionnalités et la modification de certains champs de formulaire. Au final, les modules Contacts, Projets et Tâches de projet ont été adaptés pour correspondre à la fonctionnalité Gestion des travaux.

L'outil vtigerCRM n'a quant à lui pas pu être adapté car son déploiement s'est avéré plus complexe que prévu : cette solution n'était effectivement pas prévue pour être installée sur un serveur mutualisé tel que proposé pour le test par Axianet.ch. Il a donc été mis à disposition avec sa configuration de base.

Pour que le test des outils soit plus rapide et efficace, une séance de démonstration a été organisée dans les bureaux d'Axianet.ch afin que les collaborateurs soient directement confrontés au fonctionnement des deux outils. Cette séance a également permis d'obtenir un feedback direct des utilisateurs, ce qui s'est avéré très utile pour prendre une décision.

Les fonctionnalités de SugarCRM ont été présentées et testées en suivant le processus décrit dans le cahier des charges d'Axianet.ch. Les fonctionnalités d'administration, comme la gestion des utilisateurs particulièrement pertinente, ou la gestion des différents modules dans l'outil ont également été démontrées. L'outil vtigerCRM a quant à lui été présenté de manière plus générale.

#### **4.6.1 Feedback de l'entreprise**

Après le test, l'outil qui apparaît comme correspondre le mieux aux attentes de l'entreprise est SugarCRM en raison de son interface ergonomique plus conviviale, de la richesse de ses fonctionnalités et sa facilité d'installation et d'adaptation.

Cependant, l'entreprise émet des réserves sur l'utilisation ultérieure d'un outil open-source existant et complexe, car même s'il remplit une partie voire la totalité des fonctionnalités définies dans le cadre du travail de bachelor, ils souhaitent pouvoir implémenter les autres fonctionnalités par la suite dans le système (Statistiques, gestion du serveur téléphonique Asterisk, ...). Ils préféreraient le faire sur une solution sur-mesure dont ils maîtrisent le code et le framework (Yii, déjà utilisé en interne).

De plus, l'entreprise a déjà eu des expériences négatives avec des outils de solution open-source dont le code a été adapté par ses soins. Il était apparu des problèmes au niveau de la maintenance, car à chaque mise à jour du logiciel, les modifications effectuées n'étaient plus compatibles.

## **4.7 Choix de la solution à implémenter**

Avec l'aide de mon professeur responsable, nous avons pu clairement exposer à l'entreprise les risques liés au choix d'une solution sur mesure. En effet, le produit livré en fin de projet sera un prototype, et bien que tout soit mis en œuvre pour qu'il puisse être le plus complet possible, il

est possible qu'il ne satisfasse pas entièrement aux attentes de l'entreprise, et qu'un travail sera encore à fournir pour qu'il puisse être utilisé "en production".

Comme les collaborateurs d'Axianet.ch maîtrisent déjà le framework qui sera utilisé pour le développement de la solution sur mesure, ils se sentent plus à l'aise d'apporter des modifications à un tel outil plutôt que sur un autre outil open-source dont ils ne maîtriseraient pas le cadre.

Afin d'aider à la décision, une estimation du temps de développement des fonctionnalités a également été réalisée sous la forme d'un product backlog<sup>4</sup>. La totalité des user stories estimées représente environ 170 heures de travail. Au stade où cette estimation a été effectuée, il restait environ 150 heures disponibles sur la planification dédiées à l'implémentation de la solution. On pouvait donc espérer atteindre un taux de développement suffisamment satisfaisant dans le temps imparti.

Au vu de ce qui précède et comme le travail à l'aide du framework Yii simplifiera le développement des fonctionnalités basiques, le choix s'est donc porté sur le développement d'une solution sur mesure.

## 5 Solution sur mesure

La solution sur mesure implique un développement complet. Ce développement a été réalisé à l'aide du Framework Yii, qui est présenté dans le point 5.1 de ce chapitre. On retrouve également dans cette partie la manière dont ont été modélisées les données et la description des fonctionnalités telles qu'elles ont été implémentées dans l'application.

Grâce à un espace de test et à une base de données mis à disposition par Axianet.ch, l'application était accessible sur Internet ce qui permettait de pouvoir faire tester directement une fonctionnalité dès que celle-ci était terminée, et toujours dans une approche agile, d'obtenir un feedback direct de l'utilisateur.

---

<sup>4</sup> Une explication plus ample sur les notions de product backlog et de user story est disponible au point 6.2 de ce rapport.



Illustration 6 – Page d’accueil de l’application

L’application est disponible en ligne à l’url suivante : <http://crm.milene.ch/webcrm>. Les codes d’accès sont donnés en annexe<sup>5</sup> à ce document.

## 5.1 Framework imposé : Yii

### 5.1.1 Présentation

Yii est un framework PHP qui est destiné aux développeurs d’applications web. Il est gratuit et open-source.



Illustration 7 - Logo Yii

Son nom est un acronyme de « Yes It Is ! », qui, selon leur site officiel, est souvent la meilleure réponse aux questions que peut se poser un développeur étranger à Yii : « Is it fast? ... Is it secure? ... Is it professional? ... Is it right for my next project? ... Yes, it is! »

Yii offre de nombreuses fonctionnalités dont on peut retrouver la liste complète sur leur site Internet. La partie suivante décrit les fonctionnalités qui ont été utiles pour l’application et la manière dont elles ont été utilisées.

### 5.1.2 Fonctionnalités

#### L’architecture MVC

Yii a adopté l’architecture Modèle-Vue-Contrôleur, qui permet de séparer la logique applicative de l’interface utilisateur. Ainsi, on peut facilement modifier le contenu de l’un sans en affecter l’autre. Dans cette architecture, le modèle représente les données, la vue les éléments de

<sup>5</sup> Voir annexe 13.12

l'interface utilisateurs comme du texte ou des formulaires, et le contrôleur assure la communication entre les deux.

Le diagramme ci-dessous montre la structure statique d'une application Yii :

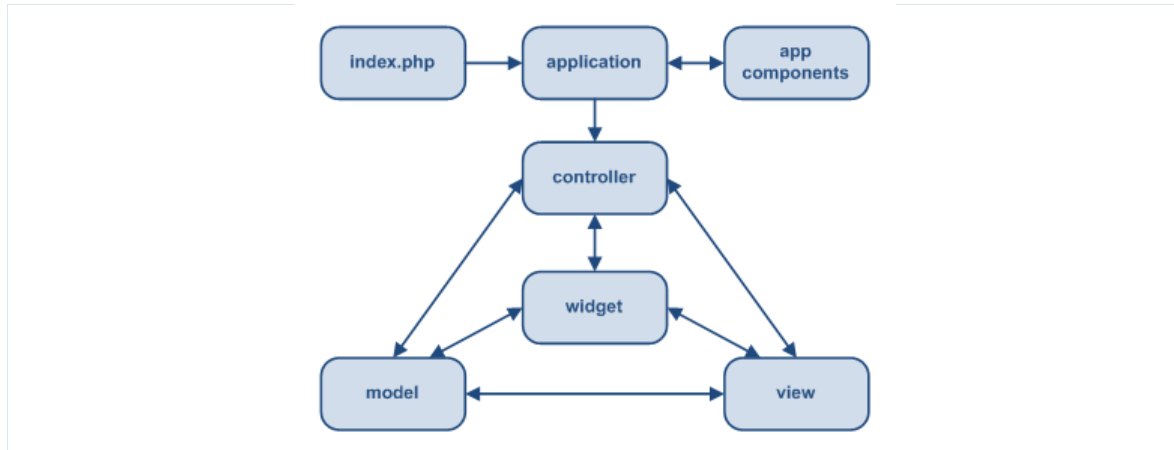


Figure 14 - Structure d'une application Yii

### Base de données : DAO, Query Builder et Active Record

Travailler avec une base de données dans le cadre de Yii est particulièrement plaisant. Tout d'abord, sa couche DAO (Data Access Objects) permet d'abstraire l'accès au système de gestion de base de données, ainsi, celui-ci peut être modifié sans modifier le code qui accède à la base de données.

Le Query Builder permet de créer des commandes SQL d'une manière orientée objet en utilisant des méthodes et des propriétés pour chaque portion de requête et les assemble en une requête SQL valide qui pourra être réutilisée en appelant l'objet.

Enfin l'Active Record est une technique populaire de mappage objet-relationnel qui donne l'illusion d'une base de données orientée objet à partir d'une base de données relationnelle. Chaque classe représente une table ou une vue de la base de données et les opérations CRUD sont implémentées en tant que méthodes de la classe. Cet exemple tiré du site internet de Yii montre comment on peut insérer une nouvelle ligne dans la table Post en utilisant l'Active Record :

```
$post=new Post;
$post->title='sample post';
$post->content='post body content';
$post->save();
```

## Gii – outil de génération de code automatique

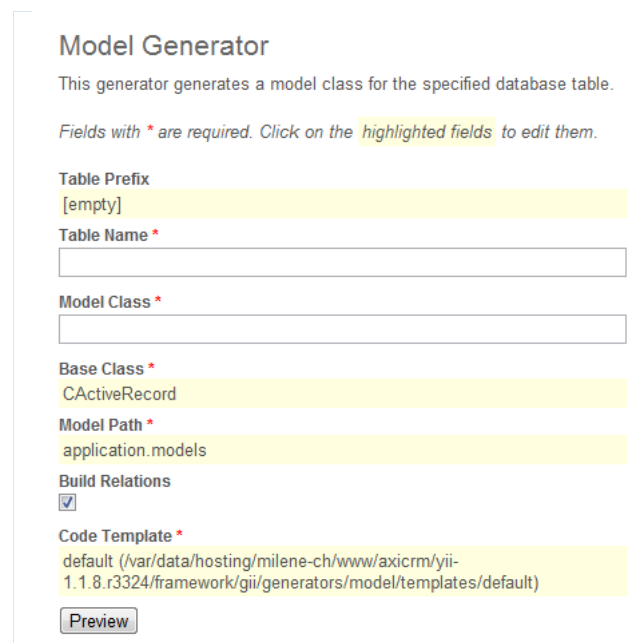
Gii est un outil, inclus dans Yii sous la forme d'un module à activer dans le fichier de configuration, qui permet de générer du code automatiquement. Gii se base sur des templates par défaut qu'il est possible de customiser.

Il permet de générer un module, un formulaire, un modèle, un contrôleur ou encore toutes les opérations CRUD (Create Read Update Delete) pour un modèle donné.

La figure ci-contre montre comment générer un modèle à partir d'une table de la base de données.

On entre tout d'abord le nom de la table, puis le nom désiré pour la classe du modèle. On peut également choisir la classe de base, l'emplacement où seront créés les fichiers, si oui ou non les relations (avec d'autres modèles) doivent être créées et le template utilisé pour la génération.

La génération de modèle et de CRUD est particulièrement efficace, car, même si les fichiers générés nécessitent évidemment d'être adaptés et configurés, elle permet de s'affranchir de la plupart des tâches répétitives et de se concentrer sur la logique de l'application.



**Model Generator**

This generator generates a model class for the specified database table.

Fields with \* are required. Click on the highlighted fields to edit them.

Table Prefix  
[empty]

Table Name \*

Model Class \*

Base Class \*  
 CActiveRecord

Model Path \*  
application.models

Build Relations  
☒

Code Template \*  
default (/var/data/hosting/milene-ch/www/axicrm/yii-1.1.8.r3324/framework/gii/generators/model/templates/default)

Preview

Figure 15 - Génération d'un modèle

## Authentification et autorisation

Yii possède un système intégré de gestion de l'authentification et des autorisations.

La partie principale du système d'authentification est un composant « Utilisateur ». Ce composant représente toutes les informations persistantes de l'utilisateur courant et peut être accédé en tout temps via `Yii::app()->user`.

Afin d'implémenter son propre système d'authentification, il convient de définir une classe « Identité » qui contient toute la logique d'authentification. Il peut en exister de nombreuses différentes afin d'implémenter différents systèmes d'authentifications (LDAP, Facebook Connect, etc.)

Pour notre application, l'implémentation de la classe d'authentification a été effectuée de manière à utiliser les informations provenant de la table utilisateurs de la base de données.

### Autres fonctionnalités

D'autres fonctionnalités utiles de Yii, comme les widgets ou la validation de formulaire ont été utilisées dans l'application.

Les widgets sont des petits composants qui permettent d'afficher des vues complexes, comme par exemple un calendrier pour sélectionner une date. L'utilisation des widgets permet une meilleure réutilisabilité du code dans les interfaces.

Un modèle possède une méthode `rules()` qui permet de définir des règles de validation pour un formulaire. Cette méthode retourne un tableau de règles de configuration. Par exemple, le code ci-dessous définit que le nom d'utilisateur est obligatoire et doit avoir une longueur minimale de 4 caractères, et une longueur maximale de 12. La 3e ligne indique que le champ password doit correspondre au champ password2.

```
array('login', 'required'),  
array('login', 'length', 'min'=>4, 'max'=>12),  
array('password', 'compare', 'compareAttribute'=>'password2'),
```

## 5.2 Modélisation des données

Le diagramme suivant représente la manière dont ont été modélisées les données pour cette application.

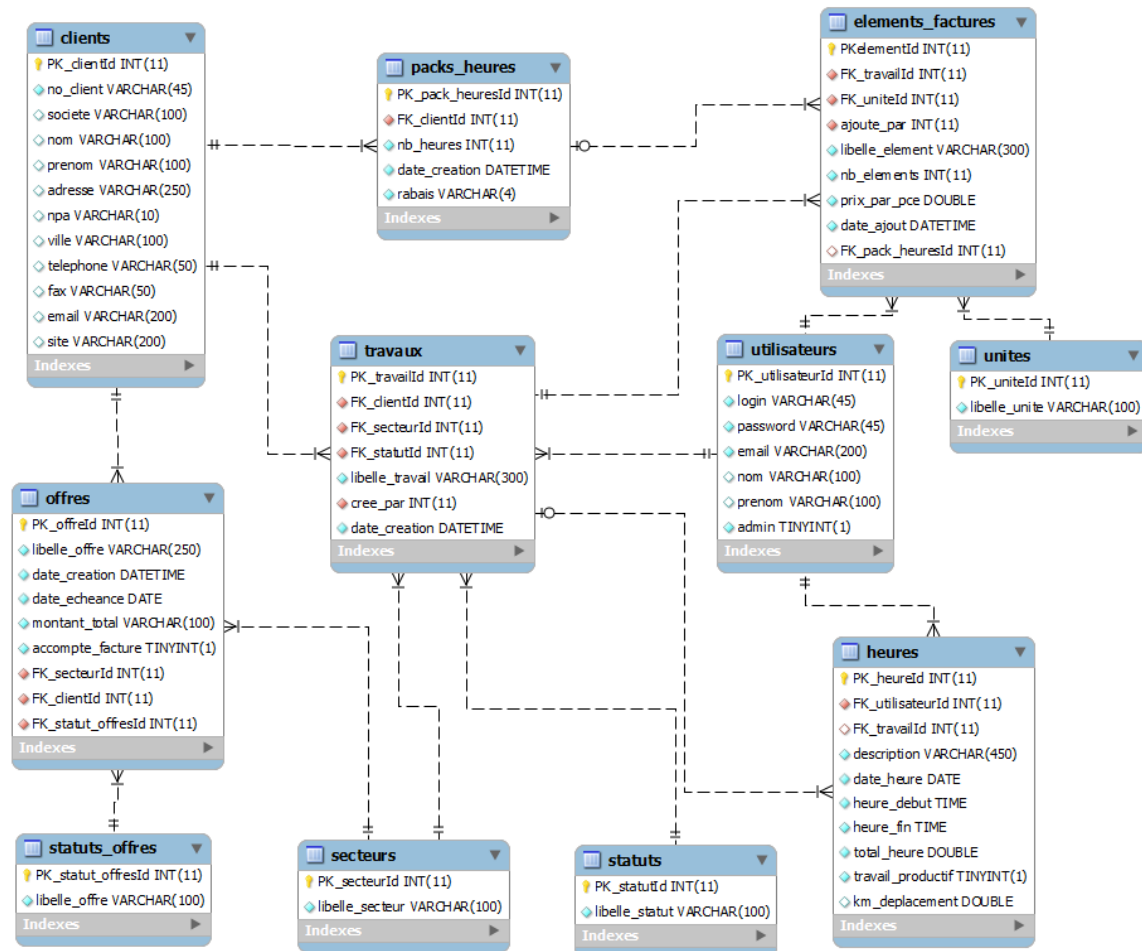


Figure 16 – Schéma de la base de données

### 5.2.1 Clients

Cette table contient toutes les informations concernant le client comme son nom, son prénom, son adresse, etc.

### 5.2.2 Offres

Tables liées : statuts\_offres, secteurs, clients

Dans le système, la gestion des offres est gérée de manière simple, elle permet d'avoir un aperçu du statut des offres et c'est pourquoi elle est reliée à la table statuts\_offres. Une offre contient entre autres une date d'échéance et le montant total de l'offre.

### 5.2.3 Statuts\_offres

Contient la liste des statuts disponibles pour une offre. Les données de base sont : Nouveau, Accepté, Relancé, Terminé.

### 5.2.4 Packs\_heures

Table liée : clients

Permet de gérer les informations liées à un pack d'heures, le nombre d'heures signifie le nombre d'heures de départ disponibles pour ce pack. Le nombre d'heures restantes n'est pas stocké dans la base de données mais calculé directement dans l'application.

### 5.2.5 Travaux

Tables liées : clients, secteurs, statuts, utilisateurs

La table travaux est l'une des tables clés de l'application. Elle permet, comme son nom l'indique, de gérer les informations d'un travail. Un travail est un projet créé pour un client, par exemple, la création d'un site internet.

### 5.2.6 Statuts

Gère le statut d'un travail : Traité, En cours, Terminé, etc.

### 5.2.7 Secteurs

Permet de gérer les secteurs d'un travail ou d'une offre : Développement web, développement mobile, etc.

### 5.2.8 Elements\_factures

Tables liées : packs\_heures, travaux, unites, utilisateurs

Un élément facturé est un élément qu'on lie à un travail pour la facturation. Celui-ci peut comporter différentes unités : des heures, des kilomètres, un forfait. Il peut également, dans le cas des heures, être déduit d'un pack d'heures que posséderait le client. On y entre le prix par pièce et le nombre d'éléments afin de connaître le total facturé. Ensuite, les totaux de tous ces éléments représenteront le montant facturé pour un travail.

### 5.2.9 Unites

Gère les unités disponibles pour un élément à facturer : Heures, Km, forfait, etc.

### 5.2.10 Utilisateurs

Un utilisateur représente un utilisateur du système, et donc un collaborateur de l'entreprise. Il possède un nom de login et un mot de passe afin de pouvoir se connecter au système.



### 5.2.11 Heures

Tables liées : utilisateurs, travaux

La table heures représente les heures effectuées par les employés. Celles-ci ne sont pas forcément facturées et peuvent correspondre à du travail non-productif (administratif, traitement des mails, etc.), elles sont par contre toujours liées à un travail pour assurer un meilleur suivi. On peut trouver dans la table des heures un champ km\_deplacement. Les informations de cette table seront consultées par la comptabilité afin de calculer les salaires et les remboursements.

## 5.3 Description de l'application

Cette partie décrit les différents modules implémentés et le mode de fonctionnement de l'application de manière globale.

### 5.3.1 Interface et navigation

Lorsqu'un utilisateur accède à l'application, il arrive sur un écran d'accueil et une page de login. Il est nécessaire de se loguer afin d'accéder aux fonctionnalités du système.



Figure 17 – Ecran d'accueil

Une fois connecté, l'utilisateur peut naviguer dans les différents modules ou se déloguer grâce au menu de navigation principal.



Figure 18 – Menu de navigation principal

Pour chaque module, un deuxième menu est disponible sur la droite de la page et permet d'effectuer des actions standards ou spécifiques au module en question, comme illustré dans la figure suivante.



Figure 19 – Menu de module

Enfin, sur chaque page, une arborescence est disponible afin d'indiquer à l'utilisateur à quel niveau de navigation il se situe.

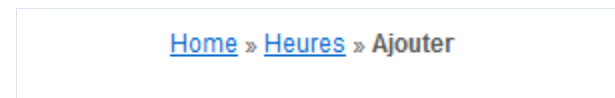


Figure 20 - Arborescence

### 5.3.2 Login

Comme mentionné dans le point précédent, il est nécessaire de se loguer afin d'accéder à l'application. Pour ce faire, un écran de login classique est disponible. La case à cocher « se souvenir » enregistre dans un cookie chez l'utilisateur ses informations (chiffrées), et le dispense ainsi de devoir entrer son login et son mot de passe à chaque connexion.

Entrez votre login et votre mot de passe :

*Les champs avec une \* sont obligatoires.*

**Username \***

**Password \***

☒ Se souvenir

Login

Figure 21 – Ecran de login

### 5.3.3 Clients

Le module client gère toutes les informations concernant les clients. Il permet de créer un client et de modifier ou d'en supprimer un existant. Sur la page d'accueil, une liste de tous les clients est disponible, depuis cette liste, on peut effectuer des tris sur toutes les colonnes, ou selon un filtre sur la valeur. On peut également procéder, pour chaque client, à quelques actions définies : voir les détails, supprimer, modifier et ajouter un travail.

Affichage 1-5 de 5 résultat(s).



















ID Client	N°	Société	Nom	Prénom	Ville	
1	235	Super SA	Super	Jean	Martigny	   
2	23	HES-SO	Dupont	Paul	Fully	   
3	23	Valcreme	Gay	Pierre		   
7	12	Marthe SA	Marthe	Sébastien		   
8	250	Téléverbier	Emery	Paul		   

Figure 22 – Gestion des clients

Lorsqu'on clique sur l'icône « voir les détails », on retrouve une page avec les informations complètes du client, et la possibilité, grâce au menu de droite, d'effectuer d'autres actions, comme ajouter un pack d'heure.

Lorsqu'une action comme l'ajout d'un travail ou d'un pack d'heure est choisie, le formulaire correspondant s'affiche avec le client déjà présélectionné.

**Actions spécifiques**

Modifier

Supprimer

Ajouter un travail

Ajouter un pack d'heure

Figure 23 - Actions spécifiques

### 5.3.4 Travaux

La page de gestion des travaux se présente relativement de la même manière que celle des clients, avec des actions accessibles pour chaque travail. Certains filtres, comme le secteur et le statut, ont des valeurs prédéfinies qui correspondent aux données configurées pour ces deux attributs.

Affichage 1-10 de 13 résultat(s).



ID travail	Client	Secteur	Statut	Libellé	Créé par	Date de création	
2	Super SA Jean Super	Hébergement		Hébergement des sites de la société	mimi	2011-07-21 14:14:28	  
3	Super SA Jean Super	Développement web		Hébergement privé www.monsite.ch	mimi	2011-07-26 16:16:39	  
4	HES-SO Paul Dupont	Hébergement	Traité	Hébergement base de données	mimi	2011-07-26 17:17:05	  
5	Marthe SA Sébastien Marthe	Développement mobile	En cours	Application iphone Marthe SA	mimi	2011-07-29 09:09:17	  
6	Super SA Jean Super	Développement web	Traité	développement CMS	mimi	2011-07-29 10:10:20	  
7	Super SA Jean Super	Développement web	En cours	Shop en ligne	mimi	2011-07-29 14:14:04	  
8	HES-SO Paul Dupont	Développement mobile	Traité	Jeu iPhone	mimi	2011-07-30 12:12:50	  

Figure 24 – Gestion des travaux

Lorsqu'on crée un travail directement depuis le client, celui-ci est déjà présélectionné dans la liste. Sinon, on peut en sélectionner un librement. On définit ensuite à quel secteur le travail appartient, et quel est son statut, et enfin on entre un libellé pour ce travail.

Client \*

HES-SO Paul Dupont

Secteur \*

Développement web

Statut \*

En cours

Libellé \*

Site avec CMS dupont.ch

Créer

Figure 25 – Création d'un travail

### 5.3.5 Éléments à facturer

Une fois un travail créé, on peut y ajouter des éléments à facturer. La gestion des éléments à facturer n'est pas accessible séparément, mais uniquement depuis un travail.

La liste des éléments à facturer dans un travail est disponible sous la vue détaillée de celui-ci. Comme nous l'avons vu pour les clients et les travaux, il est possible d'ajouter un élément à facturer directement depuis le travail où l'on se trouve.

## Application iphone Marthe SA

ID travail	5
Client	<a href="#">Marthe SA Sébastien Marthe</a>
Secteur	Développement mobile
Statut	En cours
Libellé	Application iphone Marthe SA
Créé par	mimi
Date de création	2011-07-29 09:09:17

### Éléments facturés liés

Affichage 1-3 de 3 résultat(s).







ID élément	Libellé	Nombre d'éléments	Unité	Prix par pièce (CHF)	Total (CHF)	Date d'ajout	Ajouté par	
43	Réalisation des prototypes d'écrans	16	Heures	120	1920	2011-08-11 10:10:04	mimi	 
44	Implémentation de la base de données	5	Heures	120	600	2011-08-11 10:10:04	mimi	 
45	Connexion au webservice	4	Heures	80	320	2011-08-11 10:10:05	mimi	 

Figure 26 – Affichage des éléments facturés

Il est également possible, depuis la page d'un travail, de générer un rapport imprimable contenant tous les éléments facturés et le montant total du travail afin de le faire signer au client.



## Application iphone Marthe SA

**Client** Marthe SA Sébastien Marthe  
**Libellé** Application iphone Marthe SA

### Eléments facturés liés

Libellé	Nombre d'éléments	Unité	Prix par pièce (CHF)	Total (CHF)
Réalisation des prototypes d'écrans	16	Heures	120	1920
Implémentation de la base de données	5	Heures	120	600
Connexion au webservice	4	Heures	80	320

**Montant total facturé pour ce travail : 2840 CHF**

Lieu et date

Signature

Axianet.ch Sàrl  
 Place centrale 9b-11  
 CH - 1920 Martigny  
 www.axianet.ch

Fax : +41 (0)860 79 751 55 54  
 Tél : +41 (0)27 721 88 70  
 E-mail : info@axianet.ch

Figure 27 – Rapport de travail généré

Un élément à facturer est donc évidemment lié à un travail. Il peut être de plusieurs unités différentes : heures, km, forfait, pièce. Ces unités sont configurables. On entre ensuite le nombre d'éléments qui seront facturés et le prix par élément.

Si l'unité choisie est l'heure, on a la possibilité de déduire ces heures d'un pack d'heure préenregistré, si le client en possède un. Ces packs d'heures sont en fait une quantité limitée d'heures vendues avec un rabais.

Lié au travail \*

Hébergement des sites de la société (Super SA Jean Super) ▼

Unité \*

Heures ▼

Choisissez...

Heures

Km

Pièce

Forfait

ts \*

3

Prix par pièce \*

4

Déduire du pack

☒

Déduire du pack

3 - Super SA Jean Super : 15/23h restantes (12%) ▼

Sauvegarder

Figure 28 – Ajout d'un élément à facturer

### 5.3.6 Heures

Le module Heures permet aux collaborateurs d'entrer dans le système les heures qu'ils réalisent chaque jour. Une heure est toujours liée à un travail et peut comporter un nombre de kilomètres en déplacement.

Travail lié \*

Hébergement privé www.monsite.ch (Super SA Jean Super) ▼

Description \*

séance chez le client

Date \*

2011-08-11

Heure de début \*

10:00

Heure de fin \*

11:30

Travail productif \*

☐

Km en déplacement

35

Créer

Figure 29 – Entrée d'une heure

Chaque collaborateur peut avoir un aperçu, sous forme de calendrier, du total d'heures qu'il a effectué par jour et par mois. Précisons que dans la gestion interne d'Axianet.ch, un mois de travail est considéré du 15 du mois courant au 15 du mois suivant.



Figure 30 – Consultation de ses propres heures

Ce sont sur les informations entrées dans ce module que le service de comptabilité va se baser afin de calculer les salaires et les remboursements. Celui-ci a donc accès à une page qui récapitule le total des heures et des kilomètres mensuels pour chaque collaborateur.

The screenshot shows a summary of monthly hours and kilometers for two employees. The first employee, Fauquex Milène (mimi), has a total of 12.33 hours and 50 km for the month of August 2011. The second employee, Eberle Benoit (beb), has a total of 3 hours and 0 km for the same period.

Employee	Total des heures par jour (11.08.2011)	Total des heures par mois (du 15.07.2011 au 15.08.2011)	Total des km par mois (du 15.07.2011 au 15.08.2011)
Fauquex Milène (mimi)	-	12.33	50
Eberle Benoit (beb)	-	3	-

Figure 31 – Total des heures pour la comptabilité

On retrouve également, sous chaque travail, les heures qui y sont liées.

### 5.3.7 Pack d'heures

Comme nous l'avons vu précédemment pour les éléments à facturer, il est possible de déduire des heures d'un pack d'heures.

The screenshot shows a form titled 'Deduction from pack'. It has a checked checkbox and a dropdown menu showing '2 - Valcreme Pierre Gay : 19/25h restantes (15%)'.

Figure 32 – Déduction d'un pack d'heure

The screenshot shows a form titled 'Création d'un pack d'heure'. It includes fields for 'Client' (Téléverbier Paul Emery), 'Nombre d'heures' (50), and 'Rabais' (15). There is a 'Créer' button at the bottom.

Figure 33 – Création d'un pack d'heure

On peut générer un rapport imprimable pour un pack d'heure afin de donner au client un aperçu de l'état d'utilisation de son pack.



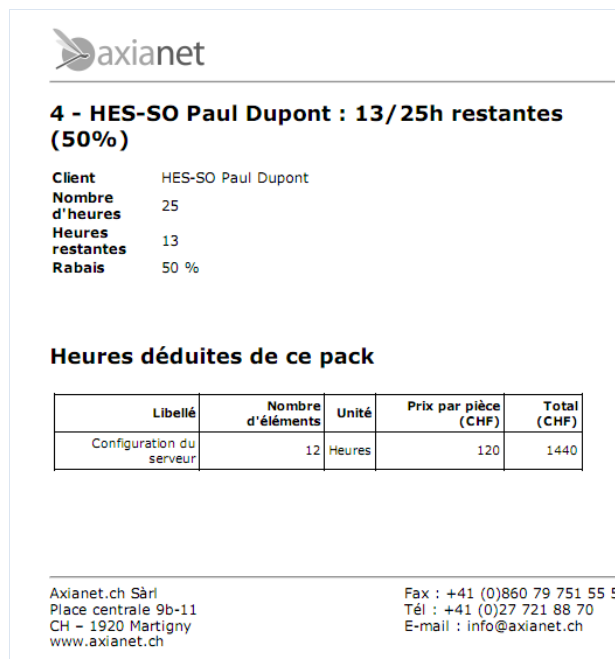


Figure 34 – Rapport d'état d'utilisation d'un pack

### 5.3.8 Offres

Le module Offres permet une gestion simple des offres en attente, avec la possibilité d'ajouter une offre pour un client, et d'ensuite gérer la liste des offres en fonction des statuts. Ces statuts sont configurables séparément.

Affichage 1-3 de 3 résultat(s).










ID offre	Libellé	Date d'échéance	Montant total	Accompte facturé	Secteur	Client	Statut de l'offre	Date de création	
1	www.super.ch	2011-08-10	20000	Oui	Développement web	Super SA Jean Super	Nouveau	2011-08-06 00:00:00	  
2	hebergement	2011-08-23	196	Non	Hébergement	HES-SO Paul Dupont	Relancé	2011-08-06 11:11:37	  
3	shop en ligne	2011-08-18	12000	Non	Développement web	Marthe SA Sébastien Marthe	Relancé	2011-08-11 10:10:18	  

Figure 35 – Gestion des offres

### 5.3.9 Utilisateurs

Le module utilisateurs permet d'ajouter, de supprimer ou de modifier des utilisateurs. C'est dans ce module que l'on peut créer des comptes d'accès pour le système.

A ce stade, tous les utilisateurs ont accès à tous les modules car l'interface permettant la gestion des rôles n'a pas été implémentée dans l'application. Des restrictions basiques peuvent

cependant être configurées manuellement dans le code de manière rapide. Chaque contrôleur possède un tableau de règles d'accès, comme l'illustre le code ci-dessous.

```
public function accessRules()
{
    return array(
        array('allow',
            'actions'=>array('create', 'edit'),
            'users'=>array('@'),
        ),
        array('deny',
            'actions'=>array('delete'),
            'users'=>array('*'),
        ),
    );
}
```

Avec ce code, on autorise la création et la modification du modèle lié à ce contrôleur aux utilisateurs enregistrés, symbolisés par le @, et on refuse la suppression aux utilisateurs non enregistrés, symbolisés par le \*. Il est également possible d'entrer directement les identifiants des utilisateurs afin d'avoir un filtre plus précis.

## 6 Gestion du projet

Afin d'assurer le bon déroulement du projet, certains mécanismes de gestion ont été mis en place. Cette section décrit quels sont ces éléments et comment a été géré le projet de manière générale. Les versions complètes des tableaux qui y sont présentés peuvent être consultées en annexe<sup>6</sup>.

### 6.1 Planification initiale

Le projet totalise 360 heures de travail sur onze semaines. Durant les cinq premières semaines, le temps de travail planifié a été de 18 heures pour s'élever à 45 heures pour les six semaines suivantes.

En début de projet, une première planification avait été réalisée jusqu'en semaine 5, à la fin de laquelle la décision quant à la solution à implémenter devait être prise. Comme nous ne possédions pas suffisamment d'éléments à ce stade pour prendre une décision, nous nous sommes accordés une semaine supplémentaire afin de laisser tester les deux outils choisis à l'entreprise. La planification a donc subi des modifications qui n'ont pas forcément pu être planifiées.

---

<sup>6</sup> Voir annexes 13.6 à 13.9

Semaines	Dates	Tâches	Prévu	Effective	Délivrables
<b>1</b>	16.05 - 22.05 (18h)	Séance de démarrage	-	2	
		Etude du cahier des charges fourni par Axianet	-	8	
		Planification séance chez Axianet	-	2	
		Préparation séance Axianet : propositions fonctionnalités, archit	-	4	
		Première recherche de CRM/ERP	-	4	
		Total	-	20	
<b>2</b>	23.05 - 29.05 (18h)	Séance Axianet	-	2	
		Rédaction PV séance	-	1	
		Analyse des besoins	-	3	
		Réalisation de la planification	-	3	Planification
		Rédaction du cahier des charges	3	3	Cahier des charges
		Recherche approfondie CRM/ERP	8	8	de CRM/ERP à analyser
		Total		20	
<b>3</b>	30.05 - 05.06 (18h)	Analyse détaillée des CRM/ERP	8	8	
		Identification des fonctionnalités	8	8	
		Comparaison des fonctionnalités	8	8	
		Total		24	
<b>4</b>	06.06 - 12.06 (18h)	Classification	4	4	
		Analyse détaillée des CRM/ERP	8	16	
		Total		20	Rapport Hebdomadaire
<b>5</b>	13.06 - 19.06 (18h)	Test des outils	8	8	
		Analyse détaillée des CRM/ERP	8	4	
		Rédaction rapport de comparaison	8	8	Rapport de comparaison
		Choix du/des CRM/ERP	-		
		Total		20	Rapport Hebdomadaire
<b>6</b>	04.07 - 10.07 (45h)	Préparation + séance proposition Axianet	-	4	
		Installation de SugarCRM et déploiement en ligne pour test	-	4	
		Installation de vTigerCRM et déploiement en ligne pour test	-	18	
		Première adaptation de SugarCRM en vue de la démo	-	14	
		Liste et estimation pour le développement des fonctionnalités n	5	5	
		Total		45	Rapport Hebdomadaire

Tableau 2 - Planification initiale

A partir de la 7e semaine, la planification ne s'est plus faite de manière globale en amont, mais semaine par semaine, sous forme d'itérations. Elles sont décrites dans le point 6.3 ci-après.

## 6.2 Approche agile

L'agilité a été présente en filigrane tout au long du projet. Si elle s'est manifestée par petites touches en début de projet, elle a pris tout son sens durant les cinq dernières semaines de travail.

En effet, dès lors que la décision fut prise de partir sur un développement sur-mesure, l'approche agile a été totalement intégrée et utilisée pour la fin du projet. La méthode agile dont

la connaissance était la plus complète était la méthode SCRUM, c'est donc sur celle-ci que notre approche s'est basée.

### 6.2.1 Product backlog

Tout d'abord, à partir du cahier des charges et des fonctionnalités décidées ensemble, un product backlog a été élaboré. Un product backlog, dans Scrum, est une liste des fonctionnalités souhaitées dans le produit. Chaque fonctionnalité a été présentée, sous la forme d'une user story. Une user story est une description, en une ou deux phrases, qui permet de cadrer ce qu'un utilisateur veut accomplir dans le système. Elle se présente typiquement dans la forme suivante :

En tant que [rôle], je veux [but/désir] afin que [bénéfice].

Chaque user story a été estimée, puis le client a également priorisé chacune d'elle selon les critères d'importances suivants : Crucial, Très haute, Haute, Normale, Faible.

Le fichier Excel, disponible en annexe<sup>7</sup>, contient la totalité du product backlog dont voici un aperçu :

#### Utilisateurs

1	En tant qu'utilisateur, je veux pouvoir me loguer et me déloguer du système.			
	Estimation	4	Priorité	Cruciale

#### Travaux (heures facturées)

17	En tant qu'utilisateur, je veux pouvoir créer un travail pour un client afin d'y entrer des éléments à facturer.			
	Estimation	5	Priorité	Cruciale

25	En tant que service de comptabilité, je veux pouvoir consulter un rapport contenant tous les éléments à facturer afin d'émettre une facture.			
	Estimation	6	Priorité	Cruciale

#### Offres

42	En tant qu'utilisateur, je veux pouvoir gérer les offres afin de pouvoir relancer les offres en attente.			
	Estimation	3	Priorité	Faible

Tableau 3 - Extrait du product backlog

<sup>7</sup> Voir annexe 13.6

### 6.2.2 Définition de « done »

La définition de « done » ou en anglais Definition of Done (DoD) est une liste de critères qu'une fonctionnalité doit remplir pour qu'elle soit considérée comme terminée. Dans le cadre de ce projet, les critères définis pour qu'une fonctionnalité soit terminée à 100% sont les suivants :

- **CRUD fonctionnel** : il faut que les opérations CRUD de base générées par Yii soient fonctionnelles
- **Validation formulaire** : il faut qu'il y ait une validation sur les champs, aussi bien les champs obligatoires que le contenu des champs
- **Champs dépendants** : les opérations CRUD telles que générées par Yii ne prennent pas en compte les champs dépendants d'autres tables, il faut que ces champs soient fonctionnels
- **Mise en page** : il faut que le formulaire soit correctement mis en page
- **Autres** : d'autres critères qui ne rentreraient pas dans les autres catégories (corrections des labels, autres...)

Dans le cas d'une itération, on cherche à livrer un prototype, ainsi pour qu'une fonctionnalité soit "done" au sens de l'itération, il faut qu'elle remplisse deux des critères ci-dessus :

- CRUD fonctionnel
- Champs dépendants

Ainsi, la fonctionnalité sera directement utilisable.

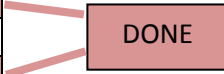
CRUD fonctionnel	20%	
Validation formulaire	20%	
Champs dépendants	20%	
Mise en page	20%	
Autres	20%	
Total	100%	

Tableau 4 – Définition de « done »

### 6.3 Itérations

Comme expliqué plus haut, dès la semaine 7, la planification s'est faite sous forme d'itérations. Il y en a eu quatre au total. Toutes les itérations ont été réalisées sur une durée d'une semaine chacune, tandis que la première itération a été un peu plus courte, sur une durée planifiée de 19h et une durée effective de 10h.

Pour chaque itération, des user stories ont été sélectionnées dans le product backlog, selon leur priorités, afin de former un iteration backlog.

Chaque fin d'itération a été signalée au client afin qu'il puisse se rendre compte de l'avancement directement sur l'application de test, disponible en ligne.

Le détail de chaque itération est disponible en annexe<sup>8</sup> sous forme de fichiers Excel. L'analyse du taux de complétion des fonctionnalités se trouve au point 7.1 de ce document.

#### 6.4 Relation avec l'entreprise

L'entreprise a bien évidemment été impliquée dans toutes les étapes du projet. Pour les points de décisions importants, des séances ont été organisées, soit dans les bureaux d'Axianet.ch, à Martigny, soit dans les locaux du Technopôle à Sierre. Au total, quatre séances de ce type ont été réalisées.

- **Une séance de démarrage :** durant cette séance, les points principaux du cahier des charges ont été repris et expliqués afin de définir, avec Axianet.ch, quelles seraient les fonctionnalités retenues dans le cadre du projet. Quelques points administratifs sur le déroulement du travail de bachelor ont également été discutés.
- **Une séance de présentation :** cette séance a servi à exposer les résultats des analyses des outils CRM/ERP existants.
- **Une séance de « conseil » :** cette séance a été organisée afin de bien informer l'entreprise quant aux conséquences liées au choix de chacune des solutions.
- **Une séance de démonstration :** durant cette séance, les deux outils précédemment choisis ont pu être démontrés à l'entreprise et testés. Cette séance a été particulièrement utile car elle a permis d'obtenir un feedback en direct des utilisateurs, ce qui a notamment aidé à prendre une décision plus rapidement.

Outre ces séances « officielles », le contact a été maintenu en permanence avec l'entreprise. Tout au long du projet, celle-ci avait accès à un dossier partagé sur Dropbox afin de pouvoir consulter en tout temps les divers documents relatifs au projet. Dans un second temps, l'application disponible en ligne sur le serveur de test a également permis de maintenir l'entreprise impliquée. De plus, les collaborateurs d'Axianet.ch ont toujours été totalement disponibles, autant par e-mail que par téléphone. Leur implication a sans aucun doute joué un rôle crucial dans le bon déroulement du projet.

---


<sup>8</sup> Voir annexe 13.7

## 7 Synthèse

Dans cette partie, on tire un bilan général à propos du déroulement du travail de bachelor. Dans le premier point, le taux de complétion de l'application est analysé. Puis, nous aborderons les principaux problèmes rencontrés durant le projet. Ensuite, nous ferons part des améliorations possibles de l'application, enfin nous amènerons un point de vue plus personnel sur le déroulement du projet.

### 7.1 Taux de complétion

Lors du développement, l'accent a été mis sur le côté fonctionnel. En effet, il était important que l'application soit tout d'abord la plus utilisable possible, c'est pourquoi la définition de « done » s'arrêtait à deux des critères établis : CRUD fonctionnel et champs dépendants.

CRUD fonctionnel	20%	
Validation formulaire	20%	
Champs dépendants	20%	
Mise en page	20%	
Autres	20%	
<b>Total</b>	<b>100%</b>	

Au final, le bilan est plutôt satisfaisant et l'on peut dire que l'état d'avancement prévu a été quelque peu sous estimé puisque la totalité des fonctionnalités ont pu être complétées à un taux de 80%, selon le tableau ci-dessous :

CRUD fonctionnel	20	%
Validation formulaire	20	%
Champs dépendants	20	%
Mise en page	10	%
Autres	10	%
<b>Total</b>	<b>80</b>	<b>%</b>

Tableau 5 – Taux de complétion des fonctionnalités

Les critères dont le taux est le plus faible, comme la mise en page ou d'autres aspects de type ergonomique seront traités dans la partie « Améliorations » de ce chapitre.

Quoi qu'il en soit le prototype livré est fonctionnel, répond aux exigences demandées et ne comprend pas de bugs ou de défaillance connus. Les points qui ont contribué à la réussite de ce projet ont été cités dans les lignes précédentes à quelques reprises : il s'agit tout d'abord de l'implication et de la disponibilité de l'entreprise, de l'approche agile mise en place pour gérer le projet et le développement à l'aide du Framework Yii a également permis de rendre l'ensemble de l'application plus cohérente.



### 7.1.1 De l'importance de l'analyse

Un autre point important qui a permis que le développement se déroule de manière plus fluide est tout le travail d'analyse effectué en amont.

En effet, on a souvent tendance à sous-estimer cette partie d'analyse et à se lancer directement dans le travail de codage. Dans le cas de ce projet, l'analyse de l'entreprise, du cahier des charges et plus particulièrement l'établissement des processus ont permis de visualiser avec précision la logique métier de l'entreprise et la manière dont l'application devait fonctionner.

De plus, l'analyse des outils existants a également permis une vision d'ensemble plus large sur les fonctionnalités à développer : les feedbacks directs des utilisateurs à propos du test de ces outils ont permis de cerner plus facilement leurs attentes. Ce processus d'analyse a donc offert une vision globale et fournit tous les outils préalables pour pouvoir se lancer dans le développement avec sérénité et précision.

## 7.2 Difficultés rencontrées

Même si globalement, le déroulement du projet s'est passé sans encombre, quelques problèmes ont toutefois surgis. Cette partie en décrit les principaux et les solutions qui y ont été apportées.

### 7.2.1 Analyse des outils

L'étude des outils a pris plus de temps que planifié car l'analyse des fonctionnalités n'a pas été aussi aisée que prévu.

En effet, certains outils étaient bien documentés et permettaient de se rendre compte facilement des propriétés qu'ils possédaient, en offrant par exemple une démonstration en ligne de la solution. D'autres en revanche se contentaient de lister les fonctionnalités présentes de manière laconique. Ces outils ont donc pour la plupart dû être téléchargés et installés en local pour être testés, et ainsi vérifier la présence des critères souhaités. Ce processus s'est révélé long et fastidieux car tous les outils ne possédaient pas forcément une procédure d'installation claire et chacun demandait une configuration différente (paramètres du serveur, base de données, etc).

Un des outils présélectionné n'a tout simplement pas pu être testé et a été retiré de l'analyse car son installation et l'accès à sa documentation se sont révélés impossibles.

La seule solution trouvée pour pallier à ce problème a été de repousser d'une semaine l'échéance pour le résultat des analyses. La planification a ainsi dû être réadaptée.

### 7.2.2 Prise de décision

Au moment de la prise de décision concernant la solution à implémenter durant la suite du travail, la situation s'est à nouveau bloquée. En effet, il était prévu qu'en consultant la grille d'analyse et lisant le rapport de comparaison, ainsi qu'en ayant un aperçu des deux outils recommandés, l'entreprise ait suffisamment d'éléments pour prendre une décision. Or ce ne fut pas le cas.

Il fallut trouver une solution rapidement afin de ne pas perdre trop de temps et ne pas prêter le développement. Avec l'aide de mon professeur responsable, nous avons organisé une séance avec un des membres d'Axianet.ch durant laquelle nous avons convenu de la marche à suivre. Il a été décidé que les deux outils recommandés seraient installés et qu'une première adaptation rapide serait faite afin que les utilisateurs puissent se rendre compte directement du fonctionnement de l'outil.

Nous avons également pu éclairer l'entreprise sur les risques liés aux choix de telle ou telle solution. Parallèlement à ça, une estimation du temps de développement des fonctionnalités a été réalisée, grâce à laquelle nous nous sommes aperçus qu'une solution sur-mesure pourrait être envisagée de manière satisfaisante.

Avec tous ces éléments supplémentaires, la décision a pu être prise rapidement et la situation s'est donc naturellement débloquée.

### **7.2.3 Appréhension du framework**

N'ayant jamais travaillé avec un framework PHP et étant plus habituée au PHP procédural qu'objet, il m'a fallu un peu de temps d'adaptation pour appréhender le framework Yii. Par ailleurs, passés les tutoriels d'initiation relativement basiques, j'ai trouvé que la documentation était assez pauvre, dans le sens que beaucoup d'articles traitaient de sujets ou de problèmes spécifiques, mais peu expliquaient le fonctionnement dans son ensemble.

Par chance, j'étais déjà familière avec l'architecture MVC utilisée dans ce framework, et une fois la matière « dégrossie », c'est devenu un vrai plaisir de développer avec ce type d'outil. A tel point qu'à l'avenir, je n'envisagerais pas un nouveau développement PHP sans framework : ceux-ci permettent un gain de temps considérable et permettent de bénéficier d'un environnement pérenne. De plus, ils offrent sur le long terme une résistance au changement particulièrement appréciable. Au final, l'utilisation d'un framework s'est avéré être un atout plutôt qu'une difficulté.

## **7.3 Améliorations**

Cette partie évoque les améliorations potentielles concernant le prototype livré. Comme il a déjà été dit, l'accent a été mis sur l'utilisabilité de l'application. Certains aspects listés ci-après ont donc été laissés de côté par manque de temps et mériteraient d'être améliorés.

### **7.3.1 Implémentation de la gestion des rôles**

Jugée de trop grande envergure, l'interface de gestion des rôles n'a pas été implémentée dans le prototype. Comme mentionné dans la dernière partie du point 5.3, les accès aux modules sont tout de même configurables manuellement dans le code, mais l'application ne possède pas d'interface visuelle permettant de gérer ces accès.

Il serait pratique d'avoir un module permettant de créer des groupes et d'attribuer des rôles afin de pouvoir gérer les accès aux différents modules de l'application. Certaines extensions externes

de Yii offrent cette possibilité, ce qui pourrait être une piste pour améliorer cet aspect là de l'application.

### **7.3.2 Amélioration de l'ergonomie**

Certains aspects ergonomiques du programme pourraient être améliorés, notamment la navigation. Actuellement, un menu liste toutes les fonctionnalités, il serait intéressant de pouvoir regrouper ces fonctionnalités par intérêts, avec par exemple des sous-onglets, ou une hiérarchie de menu.

Afin de rendre les processus plus fluides, on pourrait également envisager d'améliorer les formulaires. Tels qu'ils sont, ils permettent de sélectionner des options dans une liste déroulante, mais si l'option désirée n'existe pas encore, il faut l'ajouter depuis un menu extérieur. Il serait pratique de pouvoir ajouter ces options directement dans le formulaire, en insérant par exemple un bouton ouvrant une boîte de dialogue ou un pop-up. Pour les champs contenant de nombreuses options, comme la sélection d'un client ou d'un travail, on pourrait même envisager un système d'auto-complétion.

### **7.3.3 Amélioration de la ligne graphique**

Le thème graphique actuel de l'application est celui fournit par défaut par le framework Yii. Afin de rendre l'application plus conviviale, il serait agréable de personnaliser les fichiers CSS pour que le look général de l'application soit plus en accord avec l'identité visuelle de l'entreprise.

## **7.4 Bilan personnel**

Ce projet me tenait particulièrement à cœur puisqu'il ne faisait pas partie des sujets de travaux de bachelor disponibles initialement et que nous l'avons proposé en collaboration avec l'entreprise Axianet.ch.

Travailler avec une entreprise sur un cas concret a été un défi très enrichissant. En effet, dans ce cas, les besoins des utilisateurs sont réels et l'on sait que le résultat de notre travail sera utilisé dans des conditions réelles. C'est un défi, car il faut d'abord prendre en compte et surtout comprendre les attentes des utilisateurs, mais c'est au final ce qui rend l'expérience d'autant plus passionnante.

Ce travail m'a permis de développer mes capacités d'analyse et de communication, il m'a également appris à réagir de manière appropriée à l'imprévu et à proposer des solutions adaptées aux problématiques rencontrées. J'ai également pu mesurer l'importance du travail d'analyse dans le cadre d'un tel projet, car, même si parfois cette phase a pu apparaître quelque peu frustrante, elle a grandement facilité le travail de développement ensuite, en permettant d'acquérir une vision d'ensemble.

Pouvoir inclure les méthodologies agiles dans la gestion de ce projet m'a également plu. Sceptique au départ, j'ai pu apprécier dans ce projet la plus-value apportée par ce type de gestion, l'implication du client et la flexibilité qu'elles apportent m'a totalement convenu.

Enfin, au niveau technique, j'ai eu la chance de découvrir le framework Yii, ce qui a profondément modifier ma conception du développement web.

A mes yeux, ce travail représente de manière complète toutes les notions acquises durant ma formation et m'a offert une balance équilibrée entre les aspects techniques et business liés à notre métier. Il m'a conforté dans l'idée que l'informatique de gestion ne se résume pas à l'implémentation de technologies complexes, et que tout le processus nécessaire en amont est autant, voire plus important que le développement de l'application.

## 8 Glossaire

<b>Framework</b>	Ensemble de composants logiciels fournissant au programmeur une infrastructure sur laquelle se baser pour le développement
<b>CRUD</b>	Opérations courantes de création, de lecture, de modification et de suppression
<b>LAMP</b>	Acronyme de Linux Apache MySQL PHP.
<b>Pack d'heure</b>	Ensemble d'heures vendues à un client avec un rabais sur le prix de l'heure
<b>CRM</b>	Acronyme de Customer Relationship Management. Outil de gestion de la relation client.
<b>ERP</b>	Acronyme de Enterprise Resource Planning. Application qui coordonne l'ensemble des activités d'une entreprise
<b>Scrum</b>	Méthode agile dédiée à la gestion de projets
<b>Dropbox</b>	Service de stockage et de partage de fichiers en ligne

## 9 Bibliographie

### 9.1 Analyse des outils CRM/ERP existants

**PHILIPPE .SCOFFONI.NET** - "10 solutions de CRM open source pour les entreprises" de Philippe Scoffoni

<http://philippe.scoffoni.net/5-solutions-de-crm-open-source-pour-les-entreprises>  
dernière consultation le 30.05.2011

**INSIDECRM.COM** - "The top 10 Open-Source CRM solutions" de David Hakala

<http://www.insidecrm.com/features/top-open-source-solutions-121307/>  
dernière consultation le 30.05.2011

**PROGILIBRE.COM** - "10 ERP en open-source" de Philippe Nieuwbourg

[http://www.progilibre.com/10-ERP-en-open-source\\_a1006.html](http://www.progilibre.com/10-ERP-en-open-source_a1006.html)  
dernière consultation le 06.07.2011

**OPENERP.COM** – "OpenERP - Open Source Business Applications"

<http://www.openerp.com/>  
dernière consultation le 04.07.2011

**OPENTAPS.ORG**

<http://www.opentaps.org/>  
dernière consultation le 04.07.2011

**SUGARCRM.COM**

<http://www.sugarcrm.com/crm/fr>  
dernière consultation le 06.07.2011

**VTIGER.COM**

<http://www.vtiger.com/>  
dernière consultation le 06.07.2011

**DOLLIBAR.ORG**

<http://www.dolibarr.org/>  
dernière consultation le 04.07.2011

**COGIVEA.COM**

<http://www.cogivea.com/>  
dernière consultation le 04.07.2011

**WEBERP.ORG**

<http://www.weberp.org/HomePage>  
dernière consultation le 04.07.2011

**OPEN.MITIJA.COM** – « BlueERP »

<http://open.mitija.com/blueerp/>  
dernière consultation le 04.07.2011

## 9.2 Développement

### **YIIFRAMEWORK.COM**

<http://www.yiiframework.com/>

dernière consultation le 11.08.2011

### **YIIFRAMEWORK.COM - "The Definitive Guide to Yii"**

<http://www.yiiframework.com/doc/guide/>

dernière consultation le 11.08.2011

### **YIIFRAMEWORK.COM - "Forums Yii"**

<http://www.yiiframework.com/forum/>

dernière consultation le 06.08.2011

### **YIPLAYGROUND**

<http://www.yiplayground.cubedwater.com/>

dernière consultation le 05.08.2011

### **STACKOVERFLOW.COM**

<http://stackoverflow.com/>

dernière consultation le 06.08.2011

### **LEARN YII - "A related hyperlink column in CGridView, CDetailView, CListView" de Neil McGuigan**

<http://learnyii.blogspot.com/2011/01/related-hyperlink-column-in-cgridview.html>

dernière consultation le 31.07.2011

### **PRETTYSCRIPTS.COM - "yii : drop down list filter in CGridView"**

<http://www.prettyscripts.com/framework/yii/yii-drop-down-list-filter-in-cgridview>

dernière consultation le 03.08.2011

### **COMMENT CA MARCHE - "Apache – les fichiers .htaccess" de Jean-François Pillou et Douglas Six**

<http://www.commentcamarche.net/contents/apache/apacht.php3>

dernière consultation le 03.08.2011

## 9.3 Divers

### **OPEN SOURCE INITIATIVE - "The Open Source Definition"**

<http://www.opensource.org/docs/osd>

dernière consultation le 10.08.2011

### **WIKIPEDIA - "Scrum (méthode)"**

[http://fr.wikipedia.org/wiki/Scrum\\_\(m%C3%A9thode\)](http://fr.wikipedia.org/wiki/Scrum_(m%C3%A9thode))

dernière consultation le 10.08.2011

### **COMMENT CA MARCHE - "Enterprise Resource Planning (ERP) – Progiciel de gestion intégré" de Jean-François Pillou et Douglas Six**

<http://www.commentcamarche.net/contents/entreprise/erp.php3>

dernière consultation le 09.08.2011

**STARTUP'Z** - "Développez vos applications PHP avec un framework"

<http://www.startupz.fr/developpez-vos-applications-php-avec-un-framework>

dernière consultation le 09.08.2011

**SCRUM ALLIANCE** - "What is Definition of Done" de Dhaval Panchal

<http://www.scrumalliance.org/articles/105-what-is-definition-of-done-dod>

dernière consultation le 05.08.2011

**LOOKOUSTER.ORG** - "Le rôle du système d'information dans les entreprises d'aujourd'hui" de Sanni Lookman

<http://lookouster.org/blog/2009/06/21/le-role-du-systeme-d%E2%80%99information-dans-les-entreprises-d%E2%80%99aujourd%E2%80%99hui/>

dernière consultation le 09.08.2011

## 9.4 Source des images

**Page de garde** - <http://pmtips.net/customer-hindrance/>

**Illustration 1 - Logo d'Axianet.ch** – <http://www.axianet.ch>

**Illustration 2 – Logo open source initiative** - <http://www.cloudave.com/1042/open-source-as-a-saas-endgame-digging-a-bit-more/>

**Illustration 3 – Logo PHP** – <http://www.developpez.com>

**Illustration 4 – Logo MySQL** – <http://www.mysql.com>

**Illustration 5 – Logo SugarCRM** – <http://www.sugarcrm.com>

**Illustration 5 – Logo vtigerCRM** - <http://123ezwebs.com/blog/?paged=2>

**Illustration 7 – Logo Yii Framework** – <http://www.yiiframework.com>

**Figure 14 – Structure d'une application Yii** – <http://www.yiiframework.com>

**Figure 15 – Génération d'un modèle** - <http://www.yiiframework.com>



## 10 Table des figures

Figure 1 - Fonctionnalités initiales.....	6
Figure 2 - Fonctionnalités choisies .....	7
Figure 3 – Processus 1 : Entrer un travail (facturation).....	8
Figure 4 – Processus 2 : Entrer les heures des techniciens (non facturées) .....	9
Figure 5 – Acteurs internes du système .....	10
Figure 6 – Acteurs externes du système .....	10
Figure 7 - cas d'utilisation : utilisateurs.....	11
Figure 8 - cas d'utilisation : clients .....	11
Figure 9 - cas d'utilisation : travaux et heures .....	12
Figure 10 - cas d'utilisation « facturer élément ».....	13
Figure 11 - cas d'utilisation « Entrer les heures » .....	14
Figure 12 - cas d'utilisation : secteurs, statuts et offres.....	14
Figure 13 - Architecture.....	16
Figure 14 - Structure d'une application Yii .....	22
Figure 15 - Génération d'un modèle .....	23
Figure 16 – Schéma de la base de données .....	25
Figure 17 – Ecran d'accueil .....	27
Figure 18 – Menu de navigation principal.....	27
Figure 19 – Menu de module .....	28
Figure 20 - Arborescence.....	28
Figure 21 – Ecran de login .....	29
Figure 22 – Gestion des clients.....	29
Figure 23 - Actions spécifiques.....	29
Figure 24 – Gestion des travaux.....	30
Figure 25 – Création d'un travail.....	30
Figure 26 – Affichage des éléments facturés .....	31
Figure 27 – Rapport de travail généré.....	32
Figure 28 – Ajout d'un élément à facturer .....	32
Figure 29 – Entrée d'une heure.....	33
Figure 30 – Consultation de ses propres heures .....	33
Figure 31 – Total des heures pour la comptabilité.....	34
Figure 32 – Déduction d'un pack d'heure .....	34
Figure 33 – Création d'un pack d'heure .....	34
Figure 34 – Rapport d'état d'utilisation d'un pack.....	35
Figure 35 – Gestion des offres.....	35

## 11 Table des illustrations

Illustration 1 - Logo d'Axianet.ch.....	6
Illustration 2 – Logo open source initiative.....	15
Illustration 3 - Logo PHP .....	15
Illustration 4 - Logo MySQL .....	15
Illustration 5 - Logos SugarCRM et vtigerCRM .....	18
Illustration 6 – Page d'accueil de l'application.....	21
Illustration 7 - Logo Yii.....	21

## 12 Table des tableaux

Tableau 1- Liste des outils analysés.....	17
Tableau 2 - Planification initiale .....	38
Tableau 3 - Extrait du product backlog .....	39
Tableau 4 – Définition de « done » .....	40
Tableau 5 – Taux de complétion des fonctionnalités.....	42

## **13 Annexes**

### **13.1 Cahier des charges initial fourni par Axianet.ch**

## 13.2 Cahier des charges du projet

## **13.3 Rapport de comparaison des outils CRM/ERP existants**

## **13.4 Classification des outils CRM/ERP existants**

## 13.5 Schéma des processus

## 13.6 Product Backlog



## 13.7 Backlogs des itérations

## 13.8 Planification initiale

## 13.9 Planifications des itérations

## **13.10 Rapports hebdomadaires des semaines 4 à 10**

## **13.11 Procès-verbaux des séances**

## **13.12 Codes d'accès à l'application de test**